Titre: Title:	Optimisation stochastique de problèmes d'ordonnancement en santé
Auteur: Author:	Antoine Legrain
Date:	2015
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Legrain, A. (2015). Optimisation stochastique de problèmes d'ordonnancement en santé [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1974/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1974/
Directeurs de recherche: Advisors:	Louis-martin Rousseau, & Nadia Lahrichi
Programme: Program:	Mathématiques de l'ingénieur

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION STOCHASTIQUE DE PROBLÈMES D'ORDONNANCEMENT EN SANTÉ

ANTOINE LEGRAIN
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(MATHÉMATIQUES DE L'INGÉNIEUR)
DÉCEMBRE 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

OPTIMISATION STOCHASTIQUE DE PROBLÈMES D'ORDONNANCEMENT EN SANTÉ

présentée par : LEGRAIN Antoine
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GENDREAU Michel, Ph. D., président
M. ROUSSEAU Louis-Martin, Ph. D., membre et directeur de recherche
Mme LAHRICHI Nadia, Ph. D., membre et codirectrice de recherche
M. LODI Andrea, Doctorat ingénierie, membre
M. VAN HENTENRYCK Pascal, Ph. D., membre externe

DÉDICACE

À Dahbia

REMERCIEMENTS

Je souhaite tout d'abord remercier mes deux directeurs de recherche : Louis-Martin Rousseau et Nadia Lahrichi. Ils m'ont soutenu tout au long de ma thèse et m'ont permis de vivre d'autres expériences à travers des stages industriels et académiques. Ils m'ont aussi énormément aidé lors de la rédaction de cette thèse et de ces articles de recherche. Ce travail n'aurait jamais pu aboutir sans leurs conseils multiples et leur bienveillance.

Je voudrais remercier en particulier Marino Widmer (professeur à l'université de Fribourg en Suisse) pour son accueil chaleureux à Fribourg. Il a tout mis en œuvre pour me donner les meilleures conditions de vie et de travail lorsque que je suis arrivé en Suisse. Un grand merci à Reinhard pour sa bonne humeur permanente et à Moïra pour son accueil si chaleureux.

Merci à toute l'équipe du Centre Intégré de Cancérologie de Laval, et en particulier à Marie-Andrée Fortin (chef de département de Radio-oncologie), Fadi Hobeila (chef physicien), Bruno Carozza et Julie Héon, pour m'avoir accordé du temps et aidé à faire avancer ma thèse.

Merci à Catherine Gingras, Marc Brisson et Éric Prescott-Gagnon pour leur encadrement durant mon stage MITACS chez JDA Software. Merci aussi à Jonathan Vallée pour l'encadrement de mon deuxième stage MITACS chez AlayaCare.

J'aimerais aussi remercier les membres de mon jury pour avoir accepté d'évaluer mes travaux de thèse : Michel Gendreau, Pascal Van Hentenryck et Andrea Lodi.

Un énorme merci à tous les membres du GERAD et du CIRRELT qui ont égayé mes quatre années de thèse, et en particulier aux joueurs de tarot qui m'ont permis de me changer les idées. J'ai aussi beaucoup apprécié travailler avec Hocine Bouarab (compétition SHS), avec Jean-Bertrand et Étienne (compétition AIMMS-MOPTA) et avec Jérémy Omer et Samuel Rosat (compétition INRC-II). Une mention toute particulière pour Serge Bisaillon qui m'a beaucoup aidé et remonté le moral lorsque je codais. Merci à toute la gang du GERAD pour ses belles activités sociales : Jean-Bertrand, Jérémy, Marilène, Samuel, Sébastien et Thibault. Merci à mes collègues du CIRRELT : Aurélien, Hossein, Inès, Maria, Mehdi, Philippe et Vahid.

Enfin, je souhaite finir ces remerciements par le plus important : mon épouse Dahbia et mes parents. Ma chère Dahbia, merci infiniment pour ton soutien, ta patience, tes relectures, tes encouragements et ton amour pendant toutes ces années ! Merci à mes parents pour m'avoir toujours soutenu et encouragé à poursuivre mes études.

Je souhaite aussi mentionner tous les organismes qui m'ont soutenu financièrement pendant mon doctorat : le CRSNG, le FRQNT, l'Université de Fribourg, le CIRRELT, MITACS, JDA Software, AlayaCare et Elekta.

RÉSUMÉ

Les problèmes d'ordonnancement en santé sont complexes, car ils portent sur la fabrication d'ordonnancements qui absorbent les perturbations survenant dans le futur. Par exemple, les nouveaux patients urgents ont besoin d'être intégrés rapidement dans le planning courant. Cette thèse s'attaque à ces problèmes d'ordonnancement en santé avec de l'optimisation stochastique afin de construire des ordonnancements flexibles.

Nous étudions en premier lieu la fabrication d'horaires pour deux types d'équipes d'infirmières : l'équipe régulière qui s'occupe des unités de soins et l'équipe volante qui couvre les pénuries d'infirmières à l'hôpital. Quand les gestionnaires considèrent ce problème, soit ils utilisent une approche manuelle, soit ils investissent dans un logiciel commercial. Nous proposons une approche heuristique simple, flexible et suffisamment facile à utiliser pour être implémentée dans un tableur et qui ne requiert presque aucun investissement. Cette approche permet de simplifier le processus de fabrication et d'obtenir des horaires de grande qualité pour les infirmières. Nous présentons un modèle multi-objectif, des heuristiques, ainsi que des analyses pour comparer les performances de toutes ces méthodes. Nous montrons enfin que notre approche se compare très bien avec un logiciel commercial (CPLEX), peut être implémentée à moindre coût, et comble finalement le manque de choix entre les solutions manuelles et les logiciels commerciaux qui coûtent extrêmement cher.

Cette thèse s'attaque aussi à l'ordonnancement des chirurgies dans un bloc opératoire, fonctionnant avec un maximum de deux chirurgiens et de deux salles, en tenant compte de l'incertitude des durées d'opérations. Nous résolvons en premier lieu une version déterministe, qui utilise la programmation par contraintes, puis une version stochastique, qui encapsule le programme précédent dans un schéma de type “sample average approximation”. Ce schéma produit des plannings plus robustes qui s'adaptent mieux aux variations des durées de chirurgies.

Cette thèse présente le problème de prise de rendez-vous en temps réel dans un centre de radiothérapie. La gestion efficace d'un tel centre dépend principalement de l'optimisation de l'utilisation des machines de traitement. En collaboration avec le Centre Intégré de Cancérologie de Laval, nous faisons la planification des rendez-vous patients en tenant compte de leur priorité, du temps d'attente maximale et de la durée de traitement, le tout en intégrant l'incertitude reliée à l'arrivée des patients au centre. Nous développons une méthode hybride alliant optimisation stochastique et optimisation en temps réel pour mieux répondre aux besoins de planification du centre. Nous utilisons donc l'information des arrivées futures de patients pour dresser le portrait le plus fidèle possible de l'utilisation attendue des ressources. Des résultats sur des données réelles montrent que notre méthode dépasse les stratégies typiquement utilisées dans les centres.

Par la suite, afin de proposer un algorithme stochastique et en temps réel pour des problèmes d'allocation de ressources, nous généralisons et étendons la méthode hybride précédente. Ces problèmes sont naturellement très complexes, car un opérateur doit prendre dans un temps très limité des décisions irrévocables avec peu d'information sur les futures requêtes. Nous proposons un cadre théorique, basé sur la programmation mathématique, pour tenir compte de toutes les prévisions disponibles sur les futures requêtes et utilisant peu de temps de calcul. Nous combinons la décomposition de Benders, qui permet de mesurer l'impact futur de chaque décision, et celle de Dantzig-Wolfe, qui permet de s'attaquer à des problèmes combinatoires. Nous illustrons le processus de modélisation et démontrons l'efficacité d'un tel cadre théorique sur des données réelles pour deux applications : la prise de rendez-vous et l'ordonnancement d'un centre de radiothérapie, puis l'assignation de tâches à des employés et leur routage à travers l'entrepôt.

ABSTRACT

Scheduling problems are very challenging in healthcare as they must involve the production of plannings that absorb perturbations which arise in the future. For example, new high-priority patients needs to be quickly added in the computed plannings. This thesis tackles these scheduling problems in healthcare with stochastic optimization such as to build flexible plannings.

We first study the scheduling process for two types of nursing teams, regular teams from care units and the float team that covers for shortages in the hospital. When managers address this problem, they either use a manual approach or have to invest in expensive commercial tool. We propose a simple heuristic approach, flexible and easy enough to be implemented on spreadsheets, and requiring almost no investment. The approach leads to streamlined process and higher-quality schedules for nurses. The multi-objective model and heuristics are presented, and additional analysis is performed to compare the performance of the approach. We show that our approach compares very well with an optimization software (CPLEX solver) and may be implemented at no cost. It addresses the lack of choice between either manual solution method or a commercial package at a high cost.

This thesis tackles also the scheduling of surgical procedures in an operating theatre containing up to two operating rooms and two surgeons. We first solve a deterministic version that uses the constraint programming paradigm and then a stochastic version which embeds the former in a sample average approximation scheme. The latter produces more robust schedules that cope better with the surgeries' time variability.

This thesis presents an online appointment booking problem for a radiotherapy center. The effective management of such facility depends mainly on optimizing the use of the linear accelerators. We schedule patients on these machines taking into account their priority for treatment, the maximum waiting time before the first treatment, and the treatment duration. We collaborate with the Centre Intégré de Cancérologie de Laval to determine the best scheduling policy. Furthermore, we integrate the uncertainty related to the arrival of patients at the center. We develop a hybrid method combining stochastic optimization and online optimization to better meet the needs of central planning. We use information on the future arrivals of patients to provide an accurate picture of the expected utilization of resources. Results based on real data show that our method outperforms the policies typically used in treatment centers.

We generalize and extend the previous hybrid method to propose a general online stochastic algorithm for resource allocation problems. These problems are very difficult in their nature as one operator should take irrevocable decisions with a limited (or inexistent) information on future requests and under a very restricted computational time. We propose a mathematical programming-based framework taking advantage of all available forecasts of future requests and limited computational time. We combine Benders decomposition, which allows to measure the expected future impact of each decision, and Dantzig-Wolfe decomposition, which can tackle a wide range of combinatorial problems. We illustrate the modelling process and demonstrate the efficiency of this framework on real data sets for two applications: the appointment booking and scheduling problem in a radiotherapy center and the task assignment and routing problem in a warehouse.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

PIB	Produit Intérieur Brut
MSSS	Ministère de la Santé et des Services Sociaux
CTC	Centre de Traitement du Cancer
CICL	Centre Intégré de Cancérologie de Laval
linac	Accélérateurs Linéaires
VRP	Vehicle routing problem

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Les dépenses en santé ne cessent de croître à travers le monde. En 2013, elles représentent $9,9\%$ du produit intérieur brut (PIB) mondial et atteignent $17,1\%$ pour les États-Unis, $11,7\%$ pour la France et $10,9\%$ pour le Canada (Banque mondiale, 2013). Avec le vieillissement de la population et la croissance démographique, ces chiffres vont continuer de croître dans les prochaines années si l'efficacité des systèmes de santé n'est pas améliorée. Cette thèse propose des solutions pour améliorer la gestion du personnel infirmier, l'accès au bloc opératoire et l'accès aux services médicaux spécialisés en traitement du cancer.

Le Québec connaît un manque de personnels infirmiers depuis des années. Il faudrait entre 3000 et 4000 infirmières additionnelles pour combler la pénurie dans le réseau public (Ordre des infirmières et infirmiers du Québec, 2014). De plus, la rémunération des infirmiers au Québec représente déjà $1,5\%$ du PIB (Ministère de la Santé et des Services Sociaux du Québec, 2015). Dans ce contexte de pénurie, il est important pour chaque hôpital de savoir retenir ses infirmières. La qualité des horaires proposés aux infirmières est donc un élément important pour conserver son personnel infirmier, surtout dans un environnement de compression budgétaire.

Le bloc opératoire est aussi un pôle important de dépenses pour les hôpitaux et représente 0.2% du PIB québecois en 2007 (Ordre des infirmières et infirmiers auxiliaires du Québec, 2008). Les infrastructures, les chirurgiens ainsi que le personnel ont un coût non négligeable. De plus, les listes d'attente des patients peuvent être très longues dans certaines spécialités. Afin d'améliorer l'accès au bloc opératoire et de réduire ces coûts, les hôpitaux utilisent de plus en plus des outils provenant du génie industriel et de la recherche opérationnelle.

Enfin, le cancer est devenu depuis 2000 la première cause de mortalité au Québec (ISQ, 2012) et représente 0.4% du PIB canadien en 2013 (Comité consultatif de la Société canadienne du cancer, 2013). L'accès aux services médicaux spécialisés de qualité en oncologie et le respect des délais maximaux jugés médicalement acceptables sont ainsi une priorité pour le Ministère de la Santé et des Services Sociaux (MSSS) (Health Council Canada, 2004). Les autorités québécoises ont défini ces délais d'accès comme la période de temps écoulée entre la date où le patient est jugé médicalement prêt à subir un traitement et la date de début de ce traitement. L'objectif d'assurer l'accès sera d'autant plus une priorité que le nombre de nouveaux cas de cancer au Canada a augmenté de 12%, à un taux de 3% par année, entre 2003 et 2007 (Comité consultatif de la Société canadienne du cancer, 2013). La hausse de nouveaux cas serait attribuable à la croissance démographique et au vieillissement de la population, ce qui laisse présager que le nombre de nouveaux cas ne devrait pas cesser d'augmenter. Les autorités québécoises ont ainsi inauguré deux nouveaux centres de traitement afin de répondre à cette augmentation. Le Premier Ministre du Québec confirmait cette implication lors de l'inauguration du Centre intégré de cancérologie de la Montérégie le 21 Juin 2011 :«Cette inauguration s'inscrit dans la priorité que notre gouvernement accorde à la lutte contre le cancer. L'ouverture de ce centre et celui de Laval viennent consolider notre infrastructure de soins en oncologie»(Inauguration du centre intégré de cancérologie de Montérégie, 2011). L'amélioration de l'accès aux soins oncologiques passe naturellement par l'ouverture de nouveaux centres, mais aussi par l'optimisation de l'utilisation des ressources médicales.

Cette thèse présente des méthodes pour améliorer l'utilisation des ressources humaines et matérielles dans un contexte où l'incertitude et les coûts sont très élevés. Plus particulièrement, nous considérons trois problèmes clés du monde de la santé : la fabrication d'horaires de personnel, la prise de rendez-vous et la construction d'emploi du temps. Ces problèmes d'ordonnancement sont étudiés pour les trois secteurs de la santé présentés ci-dessus. Comme les systèmes de santé doivent répondre très rapidement aux urgences tout en assurant une prise en charge excellente de tous les patients, cette thèse veut surtout traiter l'aspect stochastique de ces problèmes. La prise en compte des incertitudes permet de s'assurer que les solutions proposées sont bien opérationnelles dans la réalité.

La gestion quotidienne des imprévus est d'autant plus importante que, par exemple, le coût moyen au Québec d'un traitement en radio-oncologie est estimé en 2008-2009 à 4470$. Ceci n'inclut que le coût d'opération, soit le coût du personnel (excluant le coût des médecins), des fournitures et d'entretien d'équipements. L'arrivée inopinée de patients palliatifs (urgents), ou au contraire l'annulation d'un rendez-vous doivent être gérées de façon à minimiser l'impact sur les opérations : la gestion des opérations doit donc intégrer un aspect stochastique. Cependant, la prise en compte de l'incertitude requiert la construction de modèles stochastiques comportant des lois de probabilité. Les processus stochastiques doivent représenter l'évolution du système de manière suffisamment réaliste pour pouvoir optimiser les décisions en se basant sur la simulation de ces modèles. Par exemple, on doit spécifier les processus d'arrivée pour l'apparition des différents types de nouveaux patients, pour les besoins de traitements de chaque patient, les durées des interventions, les annulations, la non-disponibilité des équipements ou du personnel, etc. Les éléments aléatoires à modéliser ne sont pas indépendants en général, et la modélisation de la dépendance (par exemple entre la «lourdeur» d'un cas et les autres variables aléatoires relatives à ce patient) est complexe. Les taux d'arrivée peuvent dépendre du mois ou encore du jour de la semaine. Nous ferons cette modélisation stochastique sur la base d'une analyse statistique de données historiques.

1.2 Problématique

La qualité des horaires proposés aux infirmières est un objectif prioritaire, car il encourage la stabilité de l'emploi et attire du nouveau personnel. Comme les syndicats sont très bien implantés dans les systèmes de santé québecois, les horaires doivent aussi respecter les conventions collectives. Cependant, la fabrication de ces horaires est complexe et essaie de satisfaire différentes cibles contradictoires comme la minimisation des coûts et l'équité de la charge de travail. Dans un hôpital, chaque unité gère ses propres horaires et tente de planifier un nombre minimum d'infirmières pour chaque quart de la journée et chaque jour de la semaine. Or, les hôpitaux manquent de personnel infirmier dans le contexte québécois de pénurie et font donc appel à une équipe volante pour combler les manques quotidiens d'infirmières dans chacune des unités. Celles-ci demandent un nombre d'infirmières qui, additionné à celles déjà présentes dans l'unité, permet d'assurer une excellente qualité des soins ainsi que la sécurité de tous les patients. Cette équipe volante absorbe, au niveau global de l'hôpital, les variations aléatoires de la demande en infirmières de chaque unité.

La planification et l'ordonnancement du bloc opératoire font parti des problèmes les plus étudiés en santé et sont soumis à une multitude de facteurs incertains. Les gestionnaires ont décomposé ce problème afin d'être capables de proposer une solution, c'est-à-dire un planning. Le processus de planification et d'ordonnancement est généralement décomposé en trois phases : le planning stratégique, puis tactique et enfin opérationnel. Le niveau stratégique alloue des quotas de jours d'opérations pour chaque service. Puis, le planning tactique propose un calendrier cyclique pour le bloc opératoire. Les chirurgiens ou groupes de chirurgiens se voient allouer des blocs de temps selon les quotas. Enfin, le planning opérationnel gère les horaires du personnel du bloc opératoire ainsi que l'ordonnancement et les temps de début des opérations. Plus précisément, nous étudions l'ordonnancement des opérations sur deux salles d'opération sachant que le même chirurgien opère dans les deux salles. Une opération est composée de trois phases successives : la préparation, la chirurgie et le nettoyage. Le chirurgien se doit d'être présent juste pendant la chirurgie. Les coûts d'ouverture d'une salle, du temps supplémentaire ainsi que du temps d'attente du chirurgien sont pris en compte. L'étude de ce problème permet de déterminer s'il y a un avantage à allouer deux salles à un chirurgien.

Les centres de traitement du cancer (CTC) sont habituellement des installations d'un établissement de santé, dont le rôle est d'offrir aux patients atteints de cancer et à leurs proches un ensemble de services et de traitements qui répondent à leurs besoins. Le continuum de soins dans la lutte contre le cancer implique la promotion, la prévention, le dépistage, l'investigation, le diagnostic, le traitement, le soutien, la réadaptation, la surveillance et les soins palliatifs. Plusieurs de ces services sont généralement offerts dans les CTC. Parmi les services offerts pour le volet traitement, on compte la radiothérapie. Celle-ci vise à détruire les cellules malignes par des radiations pour les empêcher de se multiplier. Il existe deux contextes d'utilisation de la radiothérapie : dans le but de guérir (radiothérapie curative), dans le but d'atténuer la souffrance du patient dont la tumeur est à un stade trop avancé (radiothérapie palliative). L'urgence du traitement, la durée, le type, le nombre et la fréquence des traitements dépendent de la condition médicale du patient et sont déterminés par l'équipe médicale. Les administrateurs des CTC font face à de nombreux défis afin d'assurer à leur clientèle l'accès, la qualité et la sécurité des services offerts. Au Québec, le MSSS maintient une liste centrale d'accès aux services spécialisés et surveille régulièrement les temps d'attente de chaque patient. En radio-oncologie par exemple, 90% des patients doivent recevoir leur premier traitement moins de quatre semaines après que le patient a été jugé médicalement prêt. En effet, il a été étudié à maintes reprises les effets néfastes qu'un délai avant le début de la radiothérapie entraîne sur l'état clinique d'un patient (Chen et al., 2008). L'impact de ces délais varie aussi fortement en fonction de la nature et de la gravité des cancers ainsi que de l'état de la maladie chez un patient. Le MSSS conseille ainsi de respecter certains délais pour traiter un patient en fonction de son état. Par exemple, les patients en soins palliatifs doivent recevoir un traitement très rapidement (en trois jours au maximum) en raison de douleurs sévères causées par leur maladie. Bien qu'il soit difficile de prévoir exactement quand ces patients se manifesteront, leur nombre n'en est pas moins imposant (jusqu'à 30% des cas de radiothérapie dans certains CTC). De plus, le processus menant au traitement d'un patient par radiothérapie est relativement complexe. Le patient doit d'abord subir une tomodensitométrie pour la planification de ses traitements. Une série de tâches complexes doit être effectuée par une équipe spécialisée (radio-oncologue, dosimétriste et physicien) avant que celui-ci ne puisse débuter ses traitements. Les traitements de radiothérapie sont administrés avec des équipements spécialisés comme les accélérateurs linéaires (linacs). Ce processus fait donc intervenir dans une séquence précise ces ressources spécialisées (tant humaines que matérielles) disponibles en quantité limitée. L'efficacité de la gestion des opérations d'un CTC repose donc sur la capacité des gestionnaires à utiliser leurs ressources à leur plein potentiel afin de s'assurer que les patients puissent obtenir leur premier rendez-vous dans les meilleurs délais.

Cette thèse est structurée de la façon suivante. Le chapitre 2 présente une revue critique de la littérature existante. Le chapitre 3 introduit la méthodologie utilisée dans chacun des articles qui suivent. Le chapitre 4 contient l'article «The nurse scheduling problem in real-life» publié dans *Journal of Medical Systems*. Le chapitre 5 contient l'article «Operating Room Management under Uncertainty» accepté dans *Constraints*. Le chapitre 6 contient l'article «Online Stochastic Optimization of Radiotherapy Patient Scheduling» publié dans *Health Care Management Science*. Le chapitre 7 contient l'article «Combining Benders and Dantzig-Wolfe Decompositions for Online Stochastic Combinatorial Optimization» soumis à *Operations Research*. Le chapitre 8 propose une discussion générale sur les quatre articles précédents. Enfin, le chapitre 9 conclut la thèse avec quelques remarques finales.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre décrit différentes méthodes pour gérer l'incertitude dans un problème d'optimisation et montre comment elles ont déjà été appliquées pour résoudre certains problèmes d'ordonnancement en santé. La revue de littérature s'articule autour de trois axes. La section 2.1 explique comment modéliser l'incertitude pour réduire la complexité des problèmes. La section 2.2 présente différents outils de programmation stochastique permettant de chercher des solutions optimales, en particulier pour les problèmes d'horaires d'infirmières et de fabrication d'emploi du temps d'un bloc opératoire. La section 2.3 propose des méthodes d'optimisation pour résoudre des problèmes dynamiques, par exemple, la prise de rendez-vous en radiothérapie.

2.1 Modélisation de l'incertitude

La modélisation de l'incertitude et son intégration sont extrêmement importantes. Considérons qu'une distribution de probabilité qui décrit le comportement aléatoire d'un problème a déjà été choisie (ce qui peut-être fait à partir de données historiques). En optimisation, la simulation de Monte-Carlo est souvent utilisée pour intégrer l'incertitude dans un modèle. L'idée principale est de réduire l'univers des possibles à un ensemble de scénarios générés par la loi de probabilité. Cette technique a l'avantage de diminuer la complexité du problème d'optimisation tout en limitant la perte d'information. L'ensemble des scénarios représentera d'autant mieux la réalité que sa taille augmente. Cette discrétisation permet d'approximer des indicateurs probabilistes (par exemple l'espérance ou la variance) et mène à la programmation stochastique qui est présentée dans la section suivante.

L'optimisation robuste (Bertsimas and Sim, 2003) permet de gérer différemment l'incertitude : elle ne prend en compte que des intervalles de variations pour chacune des variables aléatoires et cherche des solutions qui sont réalisables sur l'ensemble de ces intervalles.

Une mauvaise modélisation de l'incertitude peut mener à des solutions moins bonnes que si l'incertitude n'était pas considérée. Avramidis et al. (2004) ont développé des modèles stochastiques sophistiqués pour les centres d'appel afin de pouvoir approcher les processus d'arrivée de ces appels. Ces processus sont difficiles à modéliser, car ils ont une forte variance et la dépendance entre les périodes est très importante. Channouf and L'Ecuyer (2012) proposent un modèle simple et flexible pour ce problème. De plus, Botev et al. (2011) présentent un algorithme s'appuyant sur les méthodes de «Markov Chain Monte Carlo» et d'échantillonnage préférentiel afin de simuler les événements rares. Ces événements sont importants lorsque l'on souhaite pouvoir réagir rapidement afin de garder une certaine qualité de service. Ces modèles doivent être par la suite intégrés dans un programme stochastique (Cezik and L'Ecuyer, 2008).

2.2 La programmation stochastique

Birge and Louveaux (2011) présentent différents outils de programmation stochastique. La décomposition de Benders (Benders, 1962) a souvent été utilisée, car elle permet de résoudre des problèmes d'optimisation avec recours. Les décisions qui doivent être prises avant que l'incertitude soit révélée sont optimisées dans un problème maître ; l'impact de ces décisions est ensuite évalué dans un sous-problème différent pour chaque scenario. Ces sous-problèmes remontent finalement de l'information sur la qualité des décisions au problème maître à travers des coupes de réalisabilité et d'optimalité. Cette procédure itérative - qu'on appelle «L-Shaped» - converge vers une solution optimale. Cette méthode a montré son efficacité pour certains problèmes de routage stochastique (Gendreau et al., 1996). Cependant, si certaines contraintes ne peuvent être modélisées par un recours, elles le sont grâce à des contraintes probabilistes qui rendent le problème bien plus complexe.

Enfin, la méthode «sample average approximation» (Kleywegt et al., 2002) permet de réduire encore la complexité du problème d'optimisation stochastique. Il s'agit de résoudre le problème sur plusieurs petits ensembles de scénarios (l'ensemble peut être réduit à un seul élément) afin d'obtenir plusieurs solutions. Chacune des solutions est ensuite estimée sur un ensemble bien plus grand de scénarios (dans l'esprit de la méthode de Monte-Carlo), pour garder la meilleure d'entre elles.

2.2.1 Les horaires d'infirmières

Le problème des horaires d'infirmières est très étudié dans la littérature. Van den Bergh et al. (2013) en présentent une revue détaillée et montrent que beaucoup de méthodes ont été utilisées : recherche tabou (Burke et al., 2006), algorithme génétique (Aickelin and Dowsland, 2004), méthodes d'apprentissage automatique (Aickelin et al., 2007), «scatter search» (Burke et al., 2010) , programmation mathématique (Yilmaz, 2012) ou même des combinaisons de ces méthodes (Dowsland and Thompson, 2000; Valouxis and Housos, 2000). Burke et al. (2004) soulignent même que de meilleures solutions sont obtenues avec des méthodes modélisant précisément la problématique considérée. Cependant, certains auteurs, comme Maenhout and Vanhoucke (2006, 2007), développent des modèles et des algorithmes génériques pour le problème de fabrication d'horaires d'infirmières.

Bien que ce problème ait été beaucoup étudié, l'implémentation des solutions n'en reste pas moins problématique. Comme les horaires sont souvent construits dans un contexte de pénurie, les contraintes sur le nombre d'infirmières ne peuvent pas être respectées (Ferland et al., 2001) et sont donc relâchées. Les méthodes proposent donc souvent de minimiser le nombre d'infirmières manquantes pour chaque période de travail. Par exemple, dans (Bard and Purnomo, 2007), les contraintes sur la demande en infirmières et sur leurs préférences sont relâchées avec une heuristique basée sur une relaxation lagrangienne. Cependant, la couverture de la demande n'est pas, en général, le seul objectif considéré dans ces problèmes. Il est donc nécessaire de prioriser les différents objectifs. Berrada et al. (1996) introduisent une approche multi-objectif qui différencie les contraintes souples et dures. Ferland et al. (2001) proposent une modèle d'affectation avec un ensemble d'objectif á considérer.

Finalement, comme l'absentéisme et la demande en infirmières ne sont pas connus à l'avance, il est aussi important de prendre ces incertitudes en compte quand on dénombre les besoins en infirmières. Certains auteurs, comme Bard and Purnomo (2005a,b); Clark and Walker (2011), étudient les recours possibles comme l'utilisation d'agences d'infirmières. Cependant, l'objectif de cette thèse n'est pas de proposer de nouvelles techniques de gestion, mais de simplifier et améliorer les solutions actuelles avec des méthodes d'optimisation stochastique. Siferd and Benton (1994) étudient comment les horaires varient en fonction des prédictions de la demande en infirmières. Punnakitikashem et al. (2008) proposent un programme stochastique pour évaluer correctement dans le recours le véritable coût du manque d'infirmières. Cependant, ces techniques demandent l'utilisation de logiciels d'optimisation et sont par conséquent complexes à implémenter en réalité (Kellogg and Walczak, 2007).

2.2.2 Les emplois du temps d'un bloc opératoire

Le planning opérationnel d'un bloc opératoire a été étudié selon différents angles : Pham and Klinkert (2008) listent clairement ces différentes possibilités. Beaucoup d'heuristiques et de méta-heuristiques ont été développées. Par exemple, Dexter and Traub (2002) évaluent deux stratégies simples d'ordonnancement des opérations avec différents objectifs. Ils ordonnent les opérations de deux façons : selon la première salle de disponible, ou selon la dernière. Sier et al. (1997) présentent un recuit simulé minimisant le temps supplémentaire et le temps des opérations. Dexter et al. (1999) ont aussi étudié différentes façons d'introduire de nouvelles opérations dans un planning déjà existant. La planification opérationnelle peut aussi prendre en compte le nombre de lits disponibles dans la salle de réveil. Pham and Klinkert (2008) insistent sur le fait que la préparation du patient, son opération et son réveil doivent être coordonnés au moment de la planification afin d'assurer un calendrier des opérations réalisable. Ils utilisent la programmation en nombres entiers pour planifier ces tâches. Guinet and Chaabane (2003) proposent aussi de coordonner opérations et lits de réveil : ils construisent en premier le planning tactique puis utilisent un algorithme primal-dual pour ordonnancer les opérations. La programmation en nombres entiers et des méta-heuristiques ont souvent été utilisées pour construire le planning opérationnel ; par contre, la programmation par contraintes a été très peu utilisée. Hanset et al. (2010) l'utilisent afin de construire un planning tenant compte des affinités entre employés.

Ces plannings opérationnels subissent beaucoup de perturbations, tels des retards du personnel, des opérations qui durent plus longtemps que prévu ou l'ajout d'opérations urgentes. Afin d'assurer une certaine qualité à ce planning, il serait avantageux de considérer cette incertitude pendant le processus de planification, mais peu d'articles le font. Denton and Gupta (2003) se placent dans un contexte de prise de rendez-vous : des tâches de durée indéterminée sont à ordonnancer sur un serveur. Ils utilisent la méthode «L-Shaped» pour résoudre le problème. La fonction de recours prend en compte le retard et le temps d'attente. Denton et al. (2007) choisissent le même modèle stochastique dans le cadre d'une salle opératoire. Ils planifient le début des opérations en supposant leur ordre déjà fixé. Ensuite, ils évaluent dans le recours le temps supplémentaire, les délais d'attente des patients et le temps inutilisé d'une salle opératoire. Ils analysent par la suite l'impact de l'ordre des opérations sur le planning. Le modèle précédent est modifié pour prendre en compte les décisions d'ordonnancement. Cependant, ce modèle devient trop complexe pour être résolu. Trois heuristiques simples sont ainsi proposées pour choisir l'ordre des opérations. Les résultats montrent que de planifier en dernier les opérations qui ont le plus de variance améliore grandement le planning. De plus, ils montrent que choisir un ordonnancement quelconque réduit fortement la qualité des plannings. Van Houdenhoven et al. (2007) proposent aussi une heuristique simple afin de prendre en compte la variance sur la durée des opérations. Tout d'abord, du temps libre est cédulé à la fin de la journée pour absorber le temps supplémentaire. Ensuite, ils prennent en compte la variance du temps d'un ensemble d'opérations pour construire l'ordonnancement des salles opératoires.

2.3 L'optimisation dynamique

Dans certains problèmes d'optimisation comme la prise de rendez-vous, l'incertitude n'est pas révélée d'un seul coup, mais petit à petit. Cet aspect dynamique complexifie la résolution, car l'ensemble des possibles devient bien plus grand et plus difficile à modéliser. Les décisions doivent être prises assez rapidement et de manière répétitive : la programmation stochastique n'est pas capable en général de trouver des bonnes solutions dans le temps imparti (Powell and Roy, 2004). Deux grandes familles de méthodes permettent de résoudre ces problèmes : soit une stratégie complexe calculée à l'avance, soit une règle plus simple appliquée en temps réel.

2.3.1 Les processus de décision markoviens

Les processus de décision markoviens (Puterman, 2014) décomposent le problème en plusieurs sous-ensembles. Les différentes configurations du système sont représentées par un ensemble d'états et les décisions réalisables sont rassemblées dans un autre sous-ensemble pour chaque état. L'incertitude est modélisée par une distribution décrivant la probabilité d'atteindre un état à partir d'un autre pour chaque action. L'objectif est représenté par une fonction de transition qui donne le coût de quitter un état en fonction de l'action décidée. Une stratégie optimale est finalement calculée pour ce processus de décision markovien. Les méthodes de type «Approximate dynamic programming» (Powell, 2007) proposent de relâcher certains aspects du processus de décision pour être capable de trouver de bonnes solutions pour les problèmes complexes. Cette méthode permet de résoudre des problèmes d'optimisation financière (Bäuerle and Rieder, 2011), de réservation (Patrick et al., 2008a) et de routage (Novoa and Storer, 2009). Le désavantage majeur de cette technique réside dans le besoin d'une distribution a priori : il est tentant d'intégrer toute l'information apprise depuis le début du processus dans la loi de probabilité, ce qui n'est en général pas possible avec les processus de décision markoviens.

2.3.2 L'optimisation en temps réel

Ces techniques ont été développées afin de pouvoir prendre une décision rapidement lorsqu'un événement survient tout en gardant une bonne solution globale. Par exemple, le placement publicitaire sur les moteurs de recherche (le problème Adwords) est un problème classique. À chaque recherche d'un utilisateur, le moteur affiche des publicités associées aux mots-clés utilisés dans la recherche afin de maximiser son revenu. Les décisions doivent être prises extrêmement rapidement et assurer un revenu quasi-optimal. Un état de l'art des méthodes en temps réel est présenté dans (Jaillet and Wagner, 2010) ; le temps de résolution est le critère le plus important de ces techniques. Le type de problèmes étudiés se modélise souvent comme les problèmes de type affectation/sac à dos, ainsi que les problèmes de voyageur de commerce/tournées de véhicules, qui sont des problèmes combinatoires complexes. Ces algorithmes sont développés afin de fournir des solutions robustes quels que soient les futurs événements. Une littérature abondante et récente couvre le domaine de l'affectation. Que ce soient les problèmes de couplage (Karp et al., 1990) dans un graphe biparti ou des problèmes plus complexes comme le problème Adwords (Mehta et al., 2007), les publications sont nombreuses. L'application de l'algorithme primal-dual est particulièrement mise en avant dans (Buchbinder, 2008) où une quinzaine de problèmes différents sont explorés lorsque la modélisation mène à un problème de sac-à-dos ou d'affectation. L'algorithme primal-dual applique la même idée que l'algorithme du simplexe : à chaque itération, il fait entrer en base une variable primale qui maximise les coûts réduits. L'un des avantages de l'algorithme primal-dual est de modifier les paramètres de sélection au fur et à mesure en tenant compte des décisions passées. La difficulté pour appliquer cet algorithme réside dans la manière de mettre à jour les variables duales. Récemment, on a intégré à ces algorithmes de l'information probabiliste. Manshadi et al. (2012) déterminent une stratégie avec une distribution connue avant le début de l'algorithme pour prendre par la suite des décisions en temps réel. Par contre, peu d'articles proposent des algorithmes qui utilisent les prévisions disponibles au moment de la prise de décision. Ciocan and Farias (2012) présentent un modèle d'allocation avec des techniques de ré-optimisation pour tenir compte de cette information. Ils obtiennent d'excellents résultats pour des problèmes non combinatoires. Finalement, Van Hentenryck and Bent (2009) proposent un cadre concret pour résoudre des problèmes dynamiques avec des métaheuristiques. Pour chaque nouvelle décision à prendre, l'algorithme génère un ensemble de scénarios du futur, puis approxime sur chaque scenario l'impact de chaque décision possible. Basé sur ces approximations, l'algorithme choisit heuristiquement la meilleure décision.

2.3.3 La prise de rendez-vous en radiothérapie

L'optimisation des rendez-vous en radiothérapie est un domaine assez jeune avec une maigre revue de littérature. Le défi est de respecter les priorité des patients tout en assurant une utilisation maximale des ressources. Plusieurs stratégies simples de prise de rendez-vous en radiothérapie sont décrites dans (Petrovic et al., 2006). La stratégie «Dès que possible» (ASAP), en particulier, donne des résultats satisfaisants pour ce problème. Les patients sont triés dans l'ordre croissant des dates limites, puis, planifiés sur les équipements (dans ce cas les linacs) le plus tôt possible. D'autres approches pour construire les plannings sont proposées dans (Petrovic and Leite-Rocha, 2008). Plusieurs idées sont testées : attendre un certain temps après l'arrivée d'un patient pour lui donner un rendez-vous ou imposer des seuils d'utilisation des ressources par exemple. Ces idées ont l'avantage d'être facilement utilisables par un CTC. Dans (Kapamara et al., 2006), on peut trouver une très bonne revue de techniques plus élaborées pour résoudre ce problème : méthodes exactes, métaheuristiques, énumération implicite par exemple. Le problème y est d'ailleurs introduit comme un problème d'ordonnancement. Petrovic et al. (2009) utilisent un algorithme génétique pour résoudre le problème. Finalement, Conforti et al. (2008, 2009) et Jacquemin et al. (2010) présentent des techniques d'optimisation en utilisant la programmation en nombres entiers pour résoudre le problème de planification en radiothérapie. Ils résolvent ce problème par lot de patients rassemblés par journée ou par semaine ; ils ne donnent pas un rendez-vous au patient lorsque ce dernier arrive. De plus, toutes ces méthodes ne prennent pas en compte la distribution des profils des patients, ni le futur.

La difficulté en radiothérapie vient surtout de l'incertitude qui existe à propos des arrivées de patients : si cette variabilité n'a pas été prise en compte, les solutions proposées peuvent faire exploser les délais et les coûts. Erdogan et al. (2015) proposent une approche en temps réel afin de calculer le meilleur rendez-vous à l'arrivée du patient. Cet article expose un modèle de prise de rendez-vous en temps réel ; les problèmes sont complexes et de petite taille (10 patients). Cette méthode ne peut pour le moment résoudre des problèmes de grande taille, davantage réalistes. Sauré et al. (2012) proposent de les résoudre avec des processus de décision markoviens. Ils obtiennent de très bons résultats sur des instances générées aléatoirement à partir de statistiques provenant de l'agence du cancer de Colombie-Britannique, mais ne sont pas capables de tenir compte de l'information révélée depuis le début du processus. Des techniques d'optimisation en temps réel sont en effet plus adaptées, mais, à notre connaissance, n'ont encore jamais été appliquées dans le secteur de la santé.

CHAPITRE 3 ORGANISATION GÉNÉRALE DE LA THÈSE

Cette thèse propose différents problèmes d'ordonnancement et étudie l'optimisation stochastique de ceux-ci. L'objectif est de prendre en compte au maximum l'incertitude lors du processus de planification. Les méthodes proposées combinent différentes techniques présentées dans le chapitre 2 afin de produire les meilleures solutions possibles. En effet, aucune des méthodes décrites précédemment ne permet de résoudre les problèmes d'ordonnancement étudiés en tenant compte de toutes les prévisions disponibles.

Le chapitre 4 propose une heuristique efficace et simple à implémenter dans un tableur pour construire des horaires d'infirmières. Deux méthodes sont proposées : la première construit de manière déterministe un horaire pour chacune des unités d'un hôpital, alors que la deuxième fabrique de manière stochastique un horaire pour l'équipe volante. Cette équipe permet à l'hôpital d'absorber globalement les besoins aléatoires et supplémentaires en infirmières de chaque unité. Il est donc nécessaire de tenir compte de l'incertitude lors de la construction de l'horaire de l'équipe volante. Ces algorithmes ont été testés avec des données des hôpitaux Notre-Dame et Sainte-Justine à Montréal. Les résultats montrent qu'un bon compromis est trouvé entre qualité de la solution et complexité des algorithmes. Ce travail est présenté dans l'article «The nurse scheduling problem in real-life» publié dans *Journal of Medical Systems*. Il a reçu le prix du meilleur papier étudiant à la conférence annuelle de la *Society for Health Systems* en 2012.

Le chapitre 5 montre le travail qui a reçu une mention d'honneur à la compétition AIMMS-MOPTA 2013 Challenge (2013). Un algorithme de programmation par contraintes construit de manière déterministe un planning pour un chirurgien opérant sur deux salles : il obtient des solutions optimales très rapidement. Comme les temps d'opération sont incertains, cette procédure est intégrée dans un schéma de type «Sample Average Approximation». La programmation par contraintes génère plusieurs centaines d'emplois du temps et le schéma précédent permet d'évaluer précisément le coût moyen de chacun des plannings sur plus de mille scénarios du futur. Les résultats obtenus sur les dix instances de la compétition sont excellents et permettent de diminuer les coûts. Cependant, cette configuration est comparée à deux autres : un chirurgien avec sa salle et deux chirurgiens avec leurs salles. Cette comparaison montre que la configuration originelle est efficace, mais très instable comparée aux deux autres. Ce travail est présenté dans l'article «Operating Room Management under Uncertainty» accepté dans *Constraints*. Il a reçu le prix du meilleur papier étudiant à la *Conférence Francophone de Gestion et Ingénierie des Systèmes Hospitaliers* en 2014.

Le chapitre 6 présente le problème de prise de rendez-vous dans un centre de radiothérapie. L'efficacité de ce centre repose en premier lieu sur l'utilisation des linacs. Le centre doit assurer un accès rapide aux patients les plus urgents tout en continuant de traiter les autres dans un délai raisonnable. Ces délais sont gérés en fonction de cibles gouvernementales associées à la priorité d'un patient. Nous proposons un algorithme stochastique qui donne un rendez-vous en temps réel à chaque nouveau patient. Cet algorithme génère plusieurs prévisions sur les futurs patients du centre et propose un planning optimal pour chacune de ces prévisions. Il estime ainsi la charge de chacun des linacs et y associe un coût grâce aux variables duales. Enfin, il énumère tous les rendez-vous réalisables pour le patient et lui propose celui qui a le plus faible coût espéré. Ce coût est la somme du coût réel du rendez-vous plus le coût moyen dans le futur des plages horaires des linacs utilisées par ce rendez-vous. L'algorithme est testé sur des données réelles fournies par le Centre Intégré de Cancérologie de Laval. Les résultats obtenus sont excellents et montrent que le nombre de patients ne respectant pas les cibles gouvernementales a grandement diminué. Cet algorithme ainsi que ces résultats sont présentés dans l'article «Online Stochastic Optimization of Radiotherapy Patient Scheduling» publié dans *Health Care Management Science*. Cet article a obtenu le prix du meilleur papier étudiant à la *Conférence Francophone de Gestion et Ingénierie des Systèmes Hospitaliers* en 2012.

Enfin, le chapitre 7 propose un cadre théorique basé sur la programmation mathématique pour généraliser l'algorithme précédent et résoudre des problèmes d'allocation de ressources en temps réel. L'algorithme s'appuie sur la décomposition de Dantzig-Wolfe, qui permet de gérer des problèmes très complexes, et celle de Benders, qui donne un modèle efficace pour capter l'incertitude. Cette dernière décomposition permet à notre algorithme de calculer un coût moyen d'utilisation dans le futur de chacune des ressources et de maximiser la probabilité de respecter toutes les contraintes à la fin de l'algorithme. La nouvelle procédure est testée sur deux problèmes réels : la prise de rendez-vous et l'ordonnancement dans un centre de radiothérapie, puis l'allocation de tâches à des employés et leur routage à travers un entrepôt. La première application a été réalisée avec les données du Centre Intégré de Cancérologie de Laval (CICL). L'objectif est toujours de donner des rendez-vous aux patients, mais l'algorithme s'assure aussi qu'il y aura assez de place en dosimétrie pour finir de préparer le traitement avant la première session sur le linac. Les résultats obtenus sont similaires à ceux du chapitre précédent : le nombre de patients ne respectant pas les cibles gouvernementales demeure approximativement le même. Cependant, une grande différence se note dans le nombre de patients dont le traitement n'est pas prêt à temps : notre nouvel algorithme a permis de réduire ce chiffre de 7% à presque 0%. La deuxième application a été réalisée dans le cadre d'un stage MITACS dans l'entreprise JDA Software. Notre algorithme propose une nouvelle tâche à chaque employé dès qu'il finit la précédente. Il prend en compte la position de l'employé dans l'entrepôt, la priorité des tâches présentes dans la file d'attente et les prévisions sur les futures tâches. Les résultats, obtenus sur les données d'un client, montrent que les employés travaillent de manière plus efficace et que les tâches prioritaires sont réalisées dans le temps imparti. Ces résultats ainsi que le cadre théorique sont présentés dans l'article «Combining Benders and Dantzig-Wolfe Decompositions for Online Stochastic Combinatorial Optimization» soumis à *Operations Research*.

Les résultats de ces travaux sont discutés globalement dans le chapitre 8. Le chapitre 9 conclut finalement toute la thèse.

CHAPITRE 4 ARTICLE 1 : THE NURSE SCHEDULING PROBLEM IN REAL-LIFE

A. Legrain, H. Bouarab et N. Lahrichi ont écrit cet article et l'ont publié en 2015 dans *Journal of Medical Systems*.

4.1 Introduction

The province of Quebec is experiencing an ever-increasing demand for healthcare because of universal access and the aging of the population. However, budget cuts are unavoidable and human resources are increasingly scarce. Human resources and nursing resources in particular are the main focus of this paper. Nurses are responsible for many medical activities, and "account for approximately 25% of the total hospital operating budget and 44% of direct care costs" (Welton et al., 2006). Offering flexible schedules is a primary objective, it encourages the stability of the workforce and makes the profession more attractive in a context where there are chronic staff shortages. Providing nurses with good working conditions is also important, particularly in the Quebec context where unions are powerful and threats of strikes are often in the news.

Nurse scheduling is a complex monthly exercise with multiple and contradictory objectives such as minimizing the total costs while maximizing the satisfaction of the nurses' preferences or equally distributing the workload. The literature confirms that nurses' satisfaction is related to their workload and work conditions. In addition, the constraints imposed by collective agreements, unions, and contracts must be respected.

Constraints typically relate to :

- the quota requirement or the minimum number of nurses required to cover the units' needs for each shift ;
- the load to be assigned to each nurse in terms of the number of shifts or the category of shifts (usually defined by contract) ;
- the skill requirements on units ;
- the maximum number of consecutive days of work ;
- the minimum break between two shifts ;
- the ergonomic rules related to the number of isolated days of work or days-off for example.

Preferences are usually expressed as wished days-off or working days. In the context of nursing shortages, quota requirements should not be a "hard" constraint. This quota requirement is also referred to in the literature as a covering constraint. This constraint is the most important and challenging one. For regular units, this quota is usually defined in advance by budgets and patient-to-nurse ratio requirements. In the context of Quebec's shortages, the challenge is to be as close as possible to this quota. There is an obvious link between the regular units and the float team, since the latter absorbs shortages. The float team has available (regular) resources and also uses overtime hours, replacements, and outside on-call or agency nurses. Figure 4.1 illustrates the relationship between the two teams.

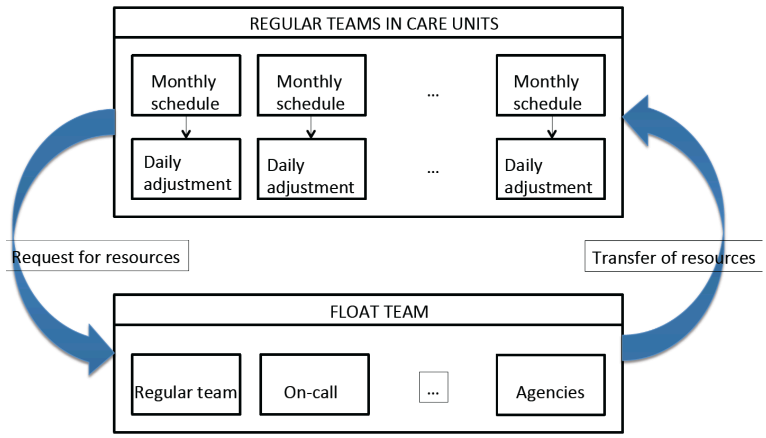


Figure 4.1 – Interaction between float team and regular care units

The adjustment of nurse schedules and determining the use of resources such as floaters and on-call nurses is discussed in (Bard and Purnomo, 2005a,b; Clark and Walker, 2011) and is not the focus of this paper. We are interested in the schedule for the regular teams and the float team.

To understand how these constraints, preferences, and workload-distribution requirements affect the scheduling process, it is essential to obtain first-hand information from a hospital unit.

Two hospitals in Montreal were contacted, Notre-Dame and Sainte-Justine. These are large public and university hospitals. The nursing structures of these hospitals are standard. There are regular nurse teams in each care unit of the hospital, supported by a centralized float team for the entire hospital, whose purpose is to absorb demand variations. The nurse scheduling process in both hospitals is decentralized, and each unit delegates this task to a scheduler responsible for manually setting up a 28-day schedule for each nurse. The workforce size in a typical unit is about 30.

Scheduling in healthcare is more often than not, manual and time-consuming and does not always provide the best results. On the other hand, commercial softwares specialized on automated scheduling may be used, but usually at a high price.

When referring to the literature on nurse rostering and scheduling, one can see that the problem is extensively studied. Van den Bergh et al. (2013) present in-depth review. They describe the methodology, models, and algorithms. A wide variety of methods have been used for nurse scheduling : mathematical programming, constraint programming, heuristics and meta-heuristics, hybrid methods, and simulation.

Different objectives are considered in the literature :

- to decrease manual scheduling ;
- to increase demand coverage in terms of workforce size and also according to required skills ;
- to maximize nurse preferences ;
- to obtain equity between the schedules.

Researchers agree that although nurse scheduling is a well-studied problem, its practical solution and the implementation at the institution are still problematic. They emphasize that better solutions are obtained when the specific features of each application are included (Burke et al., 2004). Maenhout and Vanhoucke (2006, 2007) are among the few researchers who focus on developing generic models and algorithms for the nurse scheduling problem.

Most studies are application-focused and use approaches such as tabu search (Burke et al., 2006), genetic algorithms (Aickelin and Dowsland, 2004), learning methodologies (Aickelin et al., 2007), scatter search (Burke et al., 2010) ,combinations (Dowsland and Thompson, 2000), or even mathematical programming (Yilmaz, 2012). They deal with the constraints by penalizing their violation in the objective function. It is difficult to find feasible solutions, and in numerous applications, the quota requirement constraint cannot be satisfied ; see e.g., (Ferland et al., 2001). Therefore, some researchers introduce an acceptable shortage or surplus that allows flexibility in the quota requirement. In (Bard and Purnomo, 2007), the demand constraint and respect of preferences are relaxed and a Lagrangian-based heuristic is used.

Three studies are particularly pertinent. Ferland et al. (2001) introduce an assignment-type model for the scheduling problem. They consider a set of objectives consisting of the formal objectives of the problem as well as a set of constraints. They use a tabu search (Glover and Laguna, 1997) where at each iteration, two solutions are compared by considering their objectives in a lexical order. This prioritization of objectives is central in scheduling. In (Berrada et al., 1996), a multi-objective approach is introduced that differentiates between hard and soft constraints. Valouxis and Housos (2000) formalize the nurse scheduling problem using directly the rosters (alternating between work days and rest days) in the model. A non-optimal solution is generated by solving the mathematical model and a post-optimization phase using tabu search is performed. Wong et al. (2014) solve the nurse scheduling problem in a Hong Kong emergency department with a two-phase heuristic implemented in Excel. They build a feasible planning which is then improved by a local search taking into account soft constraints. However, the nurse scheduling problem for an emergency department is a particular case as the work environment is very dynamic.

We propose solving the nurse scheduling problem for both regular and float team using a scientific method based on operations research tools, simple and easy to implement at no extra cost. Indeed, one objective is to implement our method in a spreadsheet ; nursing units already use Excel. Furthermore, as Kellogg and Walczak (2007) note, one of the reasons that even approaches based on practical studies are not implemented is the use of complicated technology. Solutions based on free softwares such as COIN-OR are therefore not suitable. We address directly this issue with our approach. Because application-based approaches are more suitable for implementation in hospitals, we focus on a specific application in the constrained context of Quebec. To summarize, our objectives are threefold : we first conduct a practical study of the process of nurse scheduling in two different large size teaching hospitals, we then introduce a procedure based on local search that can be easily implemented at no extra cost. In addition of being user-friendly, it aims for standardization and efficiency. We then conduct an analysis on the performance of the tool. We primarily focus on the practical implementation of the proposed approach rather than the optimality of the solution.

This paper is organized as follows. The next section introduces new heuristics as well a description of the transferable prototypes. The results and discussions section shows the benefits of the proposed approaches in terms of process and scheduling method and we close with concluding remarks.

4.2 Problem statement and methods

To better address the nurse scheduling problem, we first review the scheduling process and analyze non valued added tasks, and finally introduce the mathematical model and the heuristics used to solve it.

4.2.1 Analyzing the current process

In both of the hospitals that we studied, the scheduler uses three inputs to build the planning : the constraints related to work agreements, the quota requirement, and the preferences of the nurses.

Work rules We first reviewed the collective agreements to collect work rules. These rules concern the shift (or set of shifts) assigned to each person, the number of shifts per week (usually five), per two weeks, and per four weeks, the definition of fixed days if any (in some applications weekend assignments may be fixed in advance), the length of the work sequence (the number of consecutive worked days, usually limited to five), and the minimum break between two shifts (typically 16 hours). In addition to these constraints, members of the float team are, in our case, given a *typical schedule* when signing their contract. In this case, the schedule specifies when they will work (shifts and sequences) if no modifications (such as hard preferences) are requested. Since the schedule is modified every month to match the nurses' preferences, the objective is to find a compromise among the preferences without moving too far from the typical schedule. The equivalent for a regular team may be viewed as their long term (recurrent) preferences.

Quota requirement The quota requirement for a regular unit is usually set in advance and referred to as a quota. Quotas differ from day to day and from shift to shift and are fixed by considering the budget and the units' needs. In the case of regular units, at least one head nurse during the day shift is requested. Generally, quotas vary (increase or decrease) following the units' workload. An informal priority is associated with the demand for each day, with Mondays and Fridays having a higher priority because these two days appear to be critical. Shortages are prohibited on these two days.

In contrast, we typically do not plan for a quota requirement for the float team. Still a size for the core team of floaters has to be determined for each shift. The exercise of well forecasting the demand is thus essential. This demand is not known in advance since it is related to :

- the variation in the workload of the units, and
- the variation in the workforce size (for example, because of absenteeism).

In order to use a general and flexible model, we studied the historical data based on one year information to evaluate the average demand for each shift and each day (Average Demand) and the variance (Variance) for the float team. Because the quota may not realistically be met (most of the time, there is up to a 50% workforce shortage in some cases), the average demand is modified for each day of the week using the following formula :

$$(\text{Average Demand}) * (\frac{\text{Available shifts}}{\text{Total Demand}}) * (\frac{\text{Variance}}{\text{Average Variance}}),$$

where (Available shifts) is the number of shifts available in the planning period and (Total demand) is the average demand in the same period. (Average Variance) is the aggregated Variance per shift. The ratio (Available shifts / Total Demand) ensures that the total modified demand can be covered by the total available shifts over a planning period. The ratio (Variance / Average Variance) will take into account the demand variability of resources in order to schedule more nurses on high variability shifts. One should note that this ratio is easily adjustable and is used as a parameter only.

In the remaining, we will refer to this modified demand as a quota requirement for the float team.

Preferences Surprisingly, the collection of the data on preferences was not as simple as it might seem. Preferences typically relate to whether or not nurses are willing to work on a particular date. Since the schedule planning is performed every four weeks, the scheduler usually allows the nurses a few days to indicate their preferences. Each nurse will annotate the draft schedule illustrated in Figure 4.2 to indicate her preferences.

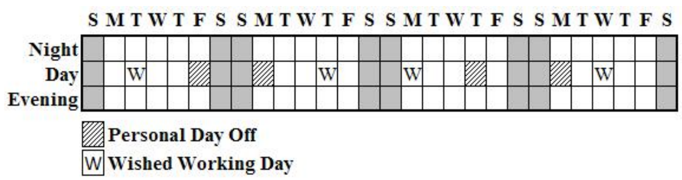


Figure 4.2 – Annotated schedule with preferences

All recurrent preferences are not mentioned because the scheduler is already aware of them. Clearly, this situation becomes very quickly problematic when the scheduler is absent or replaced.

The gathering of the preferences is interesting. The scheduler for the regular team is well organized ; she asks for preferences, and the nurses have two weeks to respond. The preferred schedule satisfies all the nurses' preferences. The scheduler for the regular floaters has no defined process : she accepts changes to the schedule throughout the planning process until the final schedule is posted. This process is time-consuming, inefficient, and certainly not optimal. We analyzed different final schedules to determine the preferences for October and November 2010 based on the number of shift combinations per nurse, the seniority, the skills, the type of rotation or shift, and the weekends.

After collecting the preferences, schedulers follow independently a three-step process starting with the sketch schedule step to create an official schedule each month. Figure 4.3 illustrates the whole process. As stated previously, all constraints relative to work rules are hard constraints and must be respected when building the schedules. Since there are shortages of the human resources, the quota requirement may be seen as a goal to reach rather than mandatory to fulfill, and particular attention is paid to Monday and Friday shifts. The weekend assignments are fixed and cannot be modified. Both teams use software, based on Excel, that is designed to manage the human resources and is linked to the payroll. This does not optimize the schedule but is used as a visual tool.

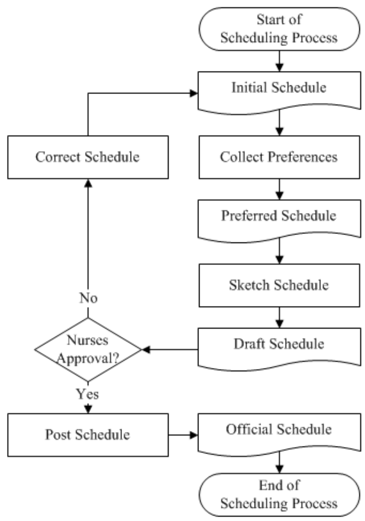


Figure 4.3 – Scheduling process map

We have identified two procedures to construct the nurse schedules in the studied hospitals : constructive or improving. The constructive procedures consists of using the draft schedule and adding shifts to reach the quota requirement. When all the mandatory shifts have been added, the following step is to try to improve the schedule one week at a time using a flip movement. Figure 4.4 illustrates this move. In the example, status of days 5 and 11, and days 18 and 21 are switched. If one was a working day, it becomes an off-day and vice versa. Allowed moves consist of moving a shift assignment from one day to another in the general case, and from one shift to another for the eligible subset of nurses.

The scheduler for the core team of the float team has an improving procedure : she moves from a completed schedule to another that better satisfies her requirements using the same flip movement. The personal judgment and experience of the scheduler are the key of choosing which schedule is best, and this is clearly difficult to model since no objective criteria are available. However, it may be summarized as reaching the quota requirement (because of the lack of resources) while satisfying the nurses' preferences. We interviewed the scheduler to recreate real situations and understand the trade-offs needed to satisfy both preferences and

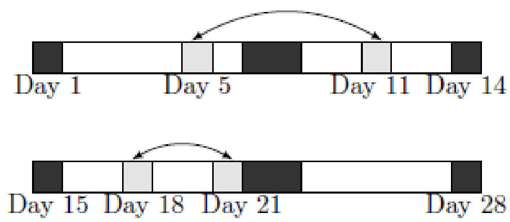


Figure 4.4 – Illustration of flip movements

quota requirements. In summary, there are no firm rules : the win-win relationship between the scheduler and the nurses often leads to informal and subjective rules that are difficult to track. The schedulers claimed that they consider the equity of different schedules, but they did not indicate how they evaluate this equity.

Once the schedule is constructed, both schedules start the "correct schedule" phase. It consists of asking for the nurses' feedback on the fulfillment of their preferences. A negotiating phase follows to convince them to accept the schedule. The schedulers then try to implement changes where possible. Finally, two weeks before the beginning of the new period, the schedule is posted in the unit. It is now final and no major changes should be made besides unpredictable daily changes (step 5 - Adjust schedule). One can see the link between regular care units and the float team who will coverage for shortages.

We carefully reviewed each step and analyzed the value added of the activities. Results and recommendations on the process are reported in the results section.

4.2.2 Solving the nurse scheduling problem

We have previously described the nurse scheduling problem and we have formally stated the mathematical model in the appendix 4.5. It considers both the regular team and the core members of the float team. Our objective is to minimize the penalties associated with ergonomics (changing shifts), quota requirement, preferences and differences from the typical schedule while ensuring a small change. This last objective is introduced to reduce resistance to change by introducing at least one difference in each period. One should note that for the float team, part of the objective function and some constraints do not apply and hence, the problem is easier to solve. Both problems are solved to optimality using the most efficient commercial optimization software package : CPLEX (IBM, 2014).

As stated earlier, CPLEX is very costly and our objective is to develop reasonable heuristics, one for each of the two teams, that are standard and simple to use, have no additional cost and resource requirements, give better results, and are less time-consuming than the manual approach. In this standardized context, the steps and rules are structured, organized, and clearly stated. Standardization leads to improvements and eases knowledge transfer in the process. It also brings objectivity on the quality of the schedule which reduces potential conflicts among staff. Furthermore, the steps are standard and simple enough to be used by a beginner. Finally, they are manual and require only pen and paper.

To facilitate their adoption, the algorithms are designed to be similar to the current working methods of the schedulers. We are using two different heuristics with different algorithms mainly to address the fact that weights of the objective function differ between the regular team scheduling problem and the float team scheduling problem. They both use the same type of movements currently used by the schedule, but in a structured manner. These algorithms are described in the appendix 4.6.

Both algorithms are implemented using Excel and if desired can be used manually. We have chosen Excel since it is simple and already used in the hospitals. Figures 4.5 and 4.6 show how the permutations are performed with Excel. A permutation is implemented by changing a value from 1 to 0 (or vice versa) : moving from an assignment to a particular shift on a particular day to no assignment (or the reverse). The schedule of each nurse is represented on a table as in Figure 4.5. Because the coverage cost on column 2 is equal to 0.3 and is negative on column 5 (respectively Monday and Thursday), the algorithm will choose a shift for a nurse whose preference is to work on Thursday rather than Monday as illustrated in the bottom table. Movements continue as long as the CONTINUE status on the left side of the table is on, as soon as it switches to STOP, the algorithm ends. The implicit stopping criteria is a local minimum related to the objective function.

4.3 Results and Discussions

4.3.1 Removing the non-added values activities

One can easily see that in practice many steps of the current process are non-value added. Two improvements are proposed, at different phases of the process, and both save time and energy.

In the sketch-schedule step, when constructing the schedule by adding shifts, the scheduler should use one spreadsheet only to avoid switching between two different, one showing the assignments and the other the quota requirements. This constant switching could be eliminated by adding three lines to count the number of nurses for each shift and day.

In the collect preferences step, the scheduler should establish a clear rule for when the list of

Continue

Demand	**11,4**	**11,9**	**10,8**	**11,1**	**13,8**	**15,0**	**12,2**	**11,4**
Modified demand	**2,0**	**1,7**	**1,8**	**1,9**	**2,7**	**3,2**	**2,2**	**2,0**
Presence	2,0	2,0	1,0	2,0	2,0	2,0	3,0	3,0
Difference	3,8	3,4	4,0	4,2	4,8	5,3	4,0	1,0
Coverage cost	0,0	0,3	-0,8	0,1	-0,7	-1,2	0,8	1,0

		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
		1	2	3	4	5	6	7	8
	Schedule		1	~~1~~			1	1	1
	Typical schedule		1	1			1	1	1
2	Days off			1					
5	Preferences		-1			1			

Figure 4.5 – Intermediate step of algorithm

STOP

Demand	**11,4**	**11,9**	**10,8**	**11,1**	**13,8**	**15,0**	**12,2**	**11,4**
Modified demand	**2,0**	**1,7**	**1,8**	**1,9**	**2,7**	**3,2**	**2,2**	**2,0**
Presence	2,0	1,0	1,0	2,0	3,0	2,0	3,0	3,0
Difference	3,8	3,4	4,0	4,2	4,8	5,3	4,0	1,0
Coverage cost	0,0	-0,7	-0,8	0,1	0,3	-1,2	0,8	1,0

		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
		1	2	3	4	5	6	7	8
	Schedule			1		1	1	1	1
	Typical schedule		1	1			1	1	1
19	Days off			1					
25	Preferences		-1			1			

Figure 4.6 – Stopping criteria of algorithm

preferences is expected and a deadline after which no more changes are accepted. Currently, the scheduler spends much time cycling between the first and third steps because she accepts a new preference, sketches the schedule, corrects it, then adds another preference, and so on. At least 50% of her time could be saved by this simple change ; the current process is time-consuming, inefficient, and certainly not optimal.

4.3.2 Comparisons of schedules quality

All the methods are compared using two main criteria : cost and time. Tables 4.1 and 4.2 summarize the results based on the following criteria. The gap from quota requirement is the under (over) coverage. The preferences are the total number of preferences respected. The equity measures the respecting of preferences across nurses ; these should be balanced. The resolution time is for the scheduler and the heuristic solution the time to manually follow all the steps ; the model column gives the CPU time consumed by the solver CPLEX. For the regular team we also compare the number of times that nurses switch shifts in a month. Finally, ergonomics indicates the quality of the schedule in terms of alternating work days and days-off. One should note that the tables do not list the objectives on the same order as the priorities for each objective depend on the scheduler or the environment.

Table 4.1 – Results for regular team

Rank	Criterion	Clerk	Heuristic	Model
1	Quota requirement	12	8	0
2	Preferences	104	112	83
3	Equity	Very high	High	High
5	Number of alternating shifts	12	15	25
6	Ergonomics	273	291	311
	Resolution time	2 to 4 days	0.5 days	0.02 s
	Cost	Low	Very high	High

Table 1 presents the results for the schedule for the regular team. The quota requirement and the preferences are contradictory objectives that are difficult to satisfy simultaneously for the scheduler, and the heuristic. The mathematical model clearly provides the best solution. The equity between the nurses is difficult to measure since the scheduler considers their long-term satisfaction. Seniority is considered only by the heuristic and the model. The scheduler pays more attention to the alternating shifts and the ergonomics ; the heuristic and the model focus more on the higher-ranking criteria. Finally, the value of the objective function is as expected, minimal with the model (-45) followed by the heuristic (160) and finally the scheduler (283). For the scheduler the construction of the schedule is time-consuming and therefore costly. The heuristic can be executed manually with no extra resources and at a low cost. In contrast, the model is solved by a commercial software CPLEX. Furthermore, even if a free solver was used, it would remain the implementation of a graphical user interface. The time reduction for providing the schedule is substantial, only half-day is necessary when using the heuristic and the solution is of better quality. Four additional quota requirements are met and preferences are better respected. The ergonomics have less weight in the objective and the heuristic, due to the preference of the scheduler. However, this heuristic is flexible enough to switch priorities.

Table 4.2 – Results for float team

Rank	Criterion	Typical Schedule	Clerk	Heuristic	Model
1	Preferences	0	None	7	12
2	Gap with the typical schedule	0	None	7	15
3	Quota requirement	7	None	5.5	5.7
4	Equity	Very high	Very high	High	High
	Resolution time	0 s	0 .5 day	30 min	0 .08 s
	Cost	Very high	Low	Very high	High

Table 4.2 presents the results for the float team. As mentioned earlier, since the process is not streamlined for this float team, no documentation on preferences with the associated final planning of a same period is available. It is referred to as "None" in the table. To provide a comparison between the different methods, we have asked for the scheduler insight. The first line is the number of respected preferences. The second line is the number of differences between the solution and the typical schedule. The quota requirement is calculated here as being the over-coverage. The equity and the cost have not been measured. Because the seniority is taken into account in the parameters, the equity is good for every solution. However, the scheduler can also check that the preferences are equitably accepted ; this second objective is not modeled in our method. This explains why the scheduler's solution is better in terms of equity. The cost of the implementation and the time consumed allow us to measure the cost. As the scheduler requires much time to build the schedule, the heuristic and the model are clearly better. The resolution time is also very important. As the global scheduling process is not completely changed, the clerk run the problem each time she receives requirements from nurses. The 0.5 days in the table shows the time spent to collect nurses' preferences. As the scheduler performs this step repeatedly, providing her

team with a schedule takes up to two weeks. Finally, our solutions would be much better if we were allowed more flexibility in the planning. A smaller emphasis on the typical schedule would introduce such flexibility. In conclusion, time reduction in planning the schedules is substantial (from half a day to 30 minutes) and the solution obtained with the heuristic is better than the manual one.

A work reorganization, improvement methods, and a standardized approach to the scheduling reduce the time required and allow both hospitals to make more efficient use of their resources. The optimization software CPLEX gives the best results, but both heuristics, considering that they are implemented in Excel, give very good results.

Additional benefits of the proposed methods should be highlighted. In our context one scheduler is assigned to each unit, and performs additional tasks to building the monthly schedule. Direct labor saving can not be observed in this context since no position can be closed. However, in other contexts where schedulers may be merged, reducing the amount of time taken to build the schedules from half a day to 30 minutes will have a direct impact in savings.

4.4 Conclusion

We have presented a practical study of nurse scheduling in medical units and for the float team in two large hospitals. The planning process is decentralized, so each scheduler develops a new schedule every month. We analyzed this process and presented a mathematical model based on multi-objective optimization for the schedule construction. We have developed simple procedures based on local search to solve the problem, which are more efficient than the manual method at no extra cost. These procedures are standard, easy to use and quick, and are based on simple permutations that can be implemented in Excel. Preferences of nurses are better respected than in the manual approach, quota requirement for each shift is closer to be achieved and generally the quality of the solutions is improved.

We are confident that our heuristic approach addresses the lack of choice between either manual solution method or a commercial package at a high cost.

4.5 Mathematical model

This section presents in detail the mathematical model.
We define the following sets.
N : set of nurses considered ;
$N_R \subset N$: set of head nurses ;

J : set of days in the period (28 days with j = 1 referring to a Sunday) ;
K : set of 8-hour shifts for each day : Night, Day and Evening.

We use the following parameters.
f_{ij} : typical schedule of nurse i for each day j : 0 (no work) and +1 (work) ;
A_i : matrix of days-off for nurse i ($a_{ij} = 0$ for work or +1 for a day-off on day j) ;
$\bar{A} = \nVdash - A$, (complementary matrix of A) ;
p_{ij} : preferences of nurse i expressed as -1 (no work), 0 (no preference), and +1 (work) for each day j (preferences are related to days, not shifts) ;
D_{jk} : quota requirement for nurses for day j and shift k ;
Q_{ik} : available shift k for nurse i (=1 if the shift is available, 0 otherwise) ;
T : maximum number of days worked by a nurse in one week ;
m_i : length of final work sequence in previous period for nurse i.

We introduce the following parameter costs.
c^+ : cost of over-covering ;
c^- : cost of under-covering ;
β_i : cost of not satisfying a preference for nurse i ;
γ_{ik} : cost of switching assignment ;
r_i : benefit from satisfying typical schedule for nurse i (aggregation of seniority, experience, skills, etc.).

To determine the most accurate value for these parameters, we have used the scheduler's insight and his personal judgment when developing the schedules to evaluate the priorities between the objectives.

The decision variables are as follows.
x_{ijk} : 1 if nurse i is assigned shift k on day j and 0 otherwise ;
z^+_{jk} : Over-coverage of day j in shift k ;
z^-_{jk} : Under-coverage of day j in shift k ;
y^+_{ijk} : auxiliary variable, positive part of $((\bar{a}_{ij}x_{ijk} - \bar{a}_{ij+1}x_{ij+1,k})$;
y^-_{ijk} : auxiliary variable, negative part of $(\bar{a}_{ij}x_{ijk} - \bar{a}_{ij+1}x_{ij+1,k})$.

We present the model using the three types of constraints : quota requirement, work rules and alternating shifts.

Quota requirement constraints

$$\sum_{i \in N_R} x_{ijDay} \geq 1, \qquad \forall j \in J \tag{4.1}$$

Constraints (4.1) ensure that at least one head nurse is present during day shifts.

$$z^+_{jk} \geq \sum_{i \in N} \bar{a}_{ij} x_{ijk} - D_{jk} \qquad \forall j \in J, \forall k \in K \tag{4.2}$$

$$z^-_{jk} \geq D_{jk} - \sum_{i \in N} \bar{a}_{ij} x_{ijk} \qquad \forall j \in J, \forall k \in K \tag{4.3}$$

Constraints (4.2) and (4.3) measure the over-coverage and the under-coverage (the gap between the quota requirement and the actual workforce) ; the days off are not taken into account ($\bar{a}_{ij} = 0$).

Work rules constraints

$$x_{ijk} \leq Q_{ik}, \qquad \forall i \in N, \forall j \in J, \forall k \in K \tag{4.4}$$

Constraints (4.4) ensure that nurses are assigned only to shifts that they are allowed to work.

$$\sum_{k \in K} x_{ijk} = 1, \qquad \forall i \in N, \forall j \in J \tag{4.5}$$

$$x_{ijDay} + x_{ij+1Night} \leq 1, \qquad \forall i \in N, \forall j = 1, ..., 27 \tag{4.6}$$

$$x_{ijEvening} + x_{ij+1Night} + x_{ij+1Day} \leq 1, \qquad \forall i \in N, \forall j = 1, ..., 27 \tag{4.7}$$

Constraints (4.5) ensure that each nurse works at most one shift per day ; constraints (4.6) and (4.7) impose a minimum break between two shifts.

$$\sum_{j=b}^{b+6} \sum_{k \in K} x_{ijk} \leq T, \qquad \forall i \in N, b = 1, 8, 15, 22 \tag{4.8}$$

$$\sum_{j=b}^{b+5} \sum_{k \in K} x_{ijk} \leq T, \qquad \forall i \in N, b = 6, ..., 23 \tag{4.9}$$

$$\sum_{j=b}^{b+5-m_i} \sum_{k \in K} x_{ijk} \leq T - m_i, \qquad \forall i \in N, \forall i \in N, b = 1, ..., 5 \tag{4.10}$$

Constraints (4.8) ensure the maximum number of work days in one week. Constraints (4.9) set the maximum number of work days to T during six consecutive days while constraints (4.10) ensure the same rules for the beginning of the month (we take into account the number of work days at the end of the last month).

$$\sum_{j=b}^{b+13} \sum_{k \in K} x_{ijk} \leq \sum_{j=b}^{b+13} f_{ij}, \qquad \forall i \in N, b = 1, 15 \tag{4.11}$$

Constraints (4.11) ensure nurses to work in two weeks exactly the same number of shifts than in their typical schedule.

$$\sum_{k \in K} x_{ijk} = f_{ij}, \qquad \forall i \in N, j = 1, 7, 8, 14, 15, 21, 22, 28 \tag{4.12}$$

Constraints (4.12) ensure they work only the weekend assigned in their typical schedule. Finally to consider days-off, constraints (4.13) set $x_{ijk} = 1$ for one shift for a day off. Even if these constraints seem to assign a shift to the nurse, we consider it to be a dummy shift to keep using the same model. This shift will can therefore be counted in the previous constraints such as the collective agreements that assign T shifts per nurse per week. One can refer to constraints (4.2) and (4.3) to understand these dummy shifts are not considered in the covering constraints.

$$\sum_{k \in K} x_{ijk} = 1, \qquad \forall (i,j) \in \{(i,j) | i \in N, j \in J, a_{ij} = 1\} \tag{4.13}$$

Alternating shifts constraints

$$y^+_{ijk} - y^-_{ijk} = \bar{a}_{ij} x_{ijk} - \bar{a}_{ij+1} x_{ij+1,k}, \qquad \forall i \in N, j = 1, ..., 27, \forall k \in K \tag{4.14}$$

Constraints (4.14) define the variables y^+_{ijk} and y^-_{ijk}. $y^+_{ijk} - y^-_{ijk}$ is equal to 1 or -1 if the nurse i does not work on the same shift the day j and $j+1$. We have added this variable since we need to minimize the number of alternating shifts.

Model

$$\begin{aligned} \min \sum_{j=1}^{27} \sum_{i \in N} \sum_{k \in K} \gamma_{ik} * (y^+_{ijk} + y^-_{ijk}) + \\ \sum_{j \in J} \sum_{k \in K} [c^+ (z^+_{jk})^2 + c^- (z^-_{jk})^2 - \sum_{i \in N} (\beta_i p_{ij} x_{ijk} - r_i f_{ij} x_{ijk})] \end{aligned} \tag{4.15}$$

subject to :

Constraints (4.1) - (4.14)

$$x_{ijk} \in \{0,1\} \qquad \forall i \in N, \forall j \in J, \forall k \in K \tag{4.16}$$

$$z^+_{jk}, z^-_{jk} \in \mathrm{R}^+ \qquad \forall j \in J, \forall k \in K \tag{4.17}$$

$$y^+_{ijk}, y^-_{ijk} \in \mathrm{R}^+ \qquad \forall i \in N, \forall j = 1, ..., 27, \forall k \in K \tag{4.18}$$

The objective function (4.15) has three terms. The first specifies that rotation from one shift to another is minimized . The second is a quadratic term that imposes a rapidly increasing penalty as the solution deviates from the quota requirements ; it will try to avoid situations where, for example, there is one over-coverage of two nurses instead of two over-coverages of one nurse. The third ensures that the preferences are maximized and equity is respected. Finally constraints (4.16), (4.17), and (4.18) ensure that the relevant variables are binary or nonnegative.

This model reflects exactly the problem for the regular team. For the float team, $\gamma_{ik} = 0$ since nurses do not rotate over shifts and constraints (4.1) is removed since no head nurse is mandatory. As there is no rotation, the model of the float team can be solved separately for each shift.

4.6 Heuristic

This section presents heuristics for the regular team and the float team problems. Algorithm 1 presents the heuristic for the regular team. The initial schedule should contain the preferences and days-off. To balance the deficit between shifts, we calculate a score using the following formula :

$$SCORE = |G_{Night} - G_{Day}| + |G_{Night} - G_{Evening}| + |G_{Day} - G_{Evening}|$$

where G_{shift} represents the monthly gap between the quota and the number of nurses assigned to a shift. Using this score, flip movements as illustrated in Figure 4.4 are performed first for the nurses eligible to shift rotation and then the set of nurses in general. Compared to the current approach used by the scheduler, the algorithm reproduces the same movements in a larger neighborhood. The flip movements are performed not in a one-week period but with the whole horizon, for all nurses with a fixed schedule (such as head nurses), and are restrained to two weeks due to a hard constraint on number of days worked every 14 days. These movements are very performant and are used in most heuristic methods when applied to large scale problems.

Algorithm 2 presents the heuristic developed for the float team. It uses permutations to minimize the objective function. A permutation $\pi^i_{j_1,j_2}$ permutes days j_1 and j_2 for nurse i. π_i

Algorithm 1 Heuristic : Regular team

INITIALIZE with the typical schedule
for each nurse **do**
  INSERT days off
  if ergonomics rules respected **then**
    INSERT preferences
  end if
end for
CALCULATE SCORE
while SCORE decreases **do**
  FIND shift s^- highest shortage **AND** shift s^+ highest surplus
  FIND week t of shift s^- highest shortage
  FIND nurse i in rotation $[s^+; s^-]$ working week t in shift s^+
  for each day in W **do**
    SWITCH from shift s^+ to shift s^- for nurse i
  end for
  CALCULATE SCORE
end while
for each shift, each week **do**
  REDUCE SHORTAGE with flip movements as illustrated in Figure 4.4
end for

represents the set of authorized permutations.

Algorithm 2 Heuristic : Float team

$x^i_j = F^i_j, \forall i \in N, \forall j = 1, ..., 28$
$\Pi cost \Leftarrow \min_{\pi^i_{j1,j2} \in \Pi} \pi cost^i_{j1,j2}$
while $\Pi cost < 0$ **do**
  FIND $(i, j1, j2)$ via $\pi cost^i_{j1,j2} = \Pi cost$
  PERMUTE x^i_{j1} and x^i_{j2}
  $\Pi cost \Leftarrow \min_{\pi^i_{j1,j2} \in \Pi} \pi cost^i_{j1,j2}$
end while

A permutation $\pi^i_{j_1,j_2}$ is authorized if it respects $x^i_{j1} + x^i_{j2} = 1$ as well as the load to be assigned (4.8)-(4.11), and is similar to the flip movement illustrated in Figure 4.4. Once again, shifts can be moved only within a two-week period, from a non-working shift to a working shift. The idea behind heuristic is to keep the same movement that the clerk is currently making. We just evaluate each movement quantitatively ; that is why our heuristic performs well on measurable objectives.

CHAPITRE 5 ARTICLE 2 : OPERATING ROOM MANAGEMENT UNDER UNCERTAINTY

JB. Gauthier et A. Legrain ont écrit cet article sous la supervision de LM. Rousseau et l'ont publié en 2015 dans *Constraints*.

5.1 Introduction

To call either of Louis Pasteur (1822-1895) or Alexander Flemming (1881-1955) the father of modern medicine would be to undermine the stellar contributions of all the other men and women of that era whom we have omitted for lack of space. Nevertheless, with the advent of more effective medical treatments and medications, life expectancy has been steadily increasing in the past two hundred years and with an even more accentuated effect in the last century. As medical treatments become more effective, dispensing of patient care no longer simply concerns the availability of medical products but also stresses hospital staff management. In other words, defying death has its cost and worldwide healthcare spending has been growing ever since. In fact, in 2012, it represents 10.2% of the global gross domestic product and even reaches 17.9% when restricting the scope to the United States of America (see World Bank, 2011). With the population ageing and the demographic density growth, there is no telling when this trend will stop but it's hard to argue that it will be any different in the coming years.

The operating block is one of the main hospital's expenditures. The infrastructures, the surgeons, and the medical staff have an important cost. Furthermore, patients waiting queues are very long for certain specialities. Hospitals and clinics thus use techniques provided by industrial engineering and operations research to improve patient access to operating rooms and reduce costs. The operating room planning and scheduling problem is consequently one of the most studied in healthcare. It is a difficult problem with a lot of uncertainties. Hospitals and clinics have decomposed it to be able to solve it. The scheduling and planning process is commonly separated in three phases : the strategic planning, the tactical planning and the operational scheduling. The strategic level decides of the quota of each type of surgeries. Then, the tactical decisions build a cyclic planning for the operating theater : the master surgical planning. According to the quotas, each surgeon or surgeon group is assigned to operating room time blocks. Finally, the operational scheduling deals with the starting time of surgeries and the staffing of the operating theatre.

Guerriero and Guido (2011) present a review of the field following this structure. Although, other authors such as Cardoen et al. (2010) and Demeulemeester et al. (2013) present literature reviews classified by characteristics, that three-level hierarchical ranking is well suited to situate a contribution. According to Guerriero and Guido (2011), the strategic planning is a resource allocation problem. The master surgical planning has often been solved with mixed-integer programming and techniques such as branch-and-price (see Beliën and Demeulemeester, 2008).

The operational scheduling has been tackled with multiple approaches taking into account different features ; Pham and Klinkert (2008) define clearly the different features as block or non-block scheduling. Many heuristics and metaheuristics have been developed. For example, Dexter and Traub (2002) evaluate two simple scheduling strategies on different objectives. They sequence surgeries according the first or the latest free operating room. Sier et al. (1997) propose a simulated annealing with a multi-objective as the overtime and the duration of surgeries. Huschka et al. (2007) use simulation to evaluate the impact of different algorithms on the patient's waiting time and the staff's overtime. They sequence the surgeries by increasing length (shortest processing time), decreasing length (longest processing time), increasing variance and increasing coefficient of variation to find that the LPT algorithm gives better results. Dexter et al. (1999) look at an existing schedule and present several algorithms to insert add-ons. M'Hallah and Al-Roomi (2014) minimize the OR overtime by using simulation to compare different online strategies such as cancelling certain surgeries. The operational scheduling also takes into account the number of free recovery beds. Pham and Klinkert (2008) insist that the coordination of the preoperative holding unit, the operating theatre, and the post-anaesthesia care unit are decisive in the feasibility of the operational scheduling. They propose a mixed integer linear program to solve this whole problem. Guinet and Chaabane (2003) build first the tactical planning with mixed-integer programming and then propose a primal-dual algorithm to schedule surgeries taking also into account free recovery beds. Hanset et al. (2010) present a constraint programming (CP) model to solve a complex scheduling problem where affinity between workers is taken into account. Saadouli et al. (2015) minimize the time that patients spend in the OR waiting for recovery beds : they solve a mixed-integer program and try to obtain robust solutions by considering the 85th-percentile instead of the average operating times.

Furthermore, all these problems contain a lot of uncertainties : the duration of surgeries, the delays of the staff, the patients no-show and the emergencies to add in the schedule. To ensure a certain quality of the operational scheduling, one must also consider these stochastic aspects. Few papers proposed stochastic formulation for solving this problem. Denton and Gupta (2003) introduce an appointment booking problem on a single resource where the duration of an appointment is stochastic. They propose a two-stage stochastic optimization formulation solved by a L-Shaped method (Birge and Louveaux, 2011). The recourse function takes into account tardiness, patient waiting time and idle time. Denton et al. (2007) use this formulation for the operating room scheduling problem. They suppose first a fixed sequence of surgeries and choose the surgery starting times with their L-Shaped method. They then propose three heuristics to sequence the surgeries before the stochastic optimization phase. The results show that scheduling surgeries with the largest variances at the end of the day improves the quality of the overall planning. Van Houdenhoven et al. (2007) also propose a heuristic to sequence surgeries with uncertain duration in one operating room. They keep a buffer time at the end of the day to absorb delays. This buffer time depends on the variance of all surgeries. Min and Yih (2010) use a Sample Average Approximation method (SAA, (Kleywegt et al., 2002)) to allocate surgeries to ORs while checking the number of free recovery beds. These approaches do not answer well to the surgeries sequencing problem, but do show that it is a complex combinatorial problem.

Scheduling one surgeon for two operating rooms (ORs) is a problem that does not seem to have been studied much before the Optimization Modeling competition of the AIMMS-MOPTA 2013 Challenge (2013) from which this paper emanates. To our knowledge, it has only been studied by Berg et al. (2009), who use discrete event simulation to analyze the patient throughput and the utilization proportion of the ORs, physicians, intake rooms and recovery rooms. They use different ratios of physicians to ORs ranging from 2-to-1 to 1-to-1.

Our paper incorporates several dimensions of the literature review. After all, *Operating Room Management under Uncertainty* such is the title of the aforementioned competition we take the opportunity to pay tribute to. The goal is to schedule one surgeon for two ORs while optimizing the quality of service within healthcare standards. In order to achieve these results, we first solve a deterministic version that uses the CP paradigm and then a stochastic version which embeds the former in a SAA scheme. We also compare our results with heuristic methods that could be used in practice.

Motivation. Berg et al. (2009) have shown that for a colonoscopy suite it would not be useful to have more than two ORs for one physician. In this spirit, we have studied three different combinations of surgeons to ORs, that is, 1-to-1, 1-to-2 and 2-to-2.

The content of this paper is as follows. Section 5.2 introduces our different models which echo the motivation of this work. Section 5.3 presents the computational results in light of each model's purpose. The paper ends with our conclusion in Section 5.4.

5.2 Optimization models

This section constitutes the core of the paper. We have yet to describe the specificities of our problem which is addressed in Section 5.2.1. Section 5.2.2 follows with some properties that are exploited by our models. We then respectively present in Sections 5.2.3 and 5.2.4 the deterministic and the stochastic models.

5.2.1 Problem specific

A surgical procedure is assumed to be decomposable in three ordered tasks : the preparation (‘P’), the actual surgery (‘S’) and the cleaning (‘C’). Let the set $\mathcal{S} := [s] \equiv \{\text{‘P’}, \text{‘S’}, \text{‘C’}\}$ capture this decomposition. Given a list of $\Sigma := [\sigma] \equiv \{1, \ldots, \mathcal{N}\}$ surgical procedures to execute during the operation hours, the goal is to determine the optimal sequence order in each of the $\mathcal{R} := [r] \equiv \{1, \ldots, \rho\}$ ORs, where $\rho \in \{1, 2\}$.

Each of these tasks requires a certain amount of time $\Pi(\sigma, s), \forall(\sigma \times s) \in \Sigma \times \mathcal{S}$ to accomplish. The sequence order entails consequences with respect to the ORs management. These consequences are decidedly measured in dollars against three factors : the vacant time (V), the waiting time (W) and the overtime (O). The first two value a wasted resource, that is respectively the OR space and the surgeon’s time. The third assumes a given regular day time period span T after which overtime salaries are paid.

A *schedule* consists of three components : sequential orderings, room assignments and time periods. That is, each surgical procedure is designated a time period thus providing a sequential order of operations in each OR. Schedules can then be compared against each other according to the cost of executing them. Optimality refers to a schedule with minimal cost.

OR vacant time corresponds to the sum of every empty time slot in the regular day time span T whereas overtime is encountered when the latter is exceeded. Finally, the surgeon’s workday is defined as the period starting at the first surgery and after the last one. A surgeon waiting time unit is accumulated for every time slot, within the workday, whereby the surgeon is not actually performing a surgery.

We underscore that for each surgery σ, the amount of time $\Pi(\sigma, s)$ required to perform task s is a random value. The methodology presented in this paper thus assumes that a random distribution for $\Pi(\sigma, s)$ is available. In this case, a sample of times for each surgical tasks is at hand (see AIMMS-MOPTA 2013 Challenge (2013)). Both the deterministic and stochastic approaches use in one way or another a realization value $\pi(\sigma, s)$. Let us call such a given set of realizations for each task a *scenario*.

5.2.2 Exploitable properties

In this section, we present properties that allow us to formulate strong models without losing track of realistic features. The first property considers the time difficulty while the second removes possibilities from the realm of feasible solutions.

The continuous nature of time is often handled by way of discretization. We retain this approach by using time units in the scale of a defined parameter τ. As such, a time length of ℓ minutes is reduced to $\lfloor \ell/\tau \rfloor$. Further observe that all time lengths in the interval $[0, \tau[$ are associated to a single time unit and so on for every other interval. In the end, when we speak of optimal solutions, it would technically be more appropriate to speak of τ-optimality. In other words, while we might miss out on some solutions that procure minute savings here and there, we believe the core problematic is addressed. Indeed, since the human factor plays a huge role, it makes little sense to plan with intractable precision.

Once the discretization is assumed, the time horizon $\Theta := [t] \equiv \{0, \ldots, \theta\}$ comes naturally in the same scale. It is also trivial to bound this domain by the total surgical procedures time $\Psi := \sum\limits_{s \in \mathcal{S}, \sigma \in \Sigma} \pi(\sigma, s)$, i.e., $\theta \leq \Psi$.

The following proposition falls outside the scope of specific models because it influences our construction on several levels. In some ways, it can also be seen as a words of wisdom with respect to the decomposition of a surgical procedure. In mathematical terms, the benefits of Proposition 1 are mostly attributed to the elimination of symmetry.

Proposition 1 *Given the current non-decreasing cost structure and the surgical team's presence throughout a surgical procedure, the tasks of the latter can be assumed to be performed back to back.*

Proof: We first show that the cleaning task should follow the surgery task straight away. This in turn eventually grants freedom to the position of the preparation task. We then show that this freedom has no impact on the evaluation of the objective function. Ultimately, Figure 5.1 illustrates one such decomposition.

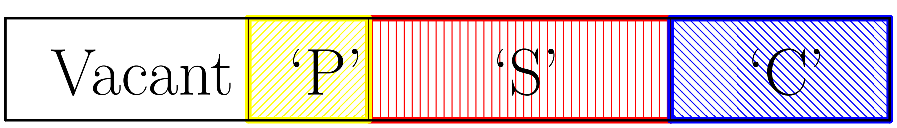


Figure 5.1 – Back to back tasks

Consider the possibilities for the cleaning task : either it is the last operation of the OR or there is another surgical procedure that follows. The conclusion for the first case is trivial. In the second case, since the surgeon is the limiting resource, the OR could be forced to incur vacant time. Given that the cost of vacant time is linear in our model, whether the vacant time is associated to said cleaning task or the preparation task that follows is irrelevant. We can therefore assume that the vacant time is condensed in the preparation period.

Given an empty time slot of length l and the preparation task of surgery σ such that $\pi(\sigma, \text{`P'}) < l$, we argue that the preparation may be accomplished anywhere within said time slot without changing the solution value. On the one hand, if the whole time slot falls in regular time, it is obvious that the cost structure ensures equivalent evaluations. On the other hand, if part of the time slot is in the overtime zone, some positions would grant a null overtime. However, we assumed that the preparation staff must stay until the surgery is over. The overtime cost is therefore the same regardless of how early the preparation is done. □

This means that if the sequence of a surgical procedure has room for interpretation, this liberty can be taken without any hesitation according to the manager's preference. The reader might want to appreciate the extreme situations such as accomplishing the preparation task as soon as possible or postponing it to the last minute.

5.2.3 Deterministic case

The deterministic model is based on a single predetermined scenario where the process time for each task corresponds to the mean time values contained in the data sample. In order to determine the optimal surgery sequence of the ORs scheduling problem, we present an optimization model that relies on the CP framework. The first model adresses both possibilities where one surgeon is available while the second model considers the 2-to-2 problem.

One surgeon

Let us introduce the activities exploited by the CP formulation as well as additional variables used to compute to objective function. The decomposition of a surgical procedure in distinctive tasks is regrouped in a cohesive design. As such, the activity $\varphi(\sigma), \forall\sigma \in \Sigma$ recuperates the natural conception and represents the combination of activities $\phi(\sigma, \text{`P'}), \phi(\sigma, \text{`S'})$ and $\phi(\sigma, \text{`C'})$ which are always consecutive in a feasible solution. The activity $\varepsilon(r), \forall r \in \mathcal{R}$ is a fictitious activity which allows the determination of the day's end for the corresponding OR. There are five other variables considered which take their values from the activities' attributes. The surgeon begin and end time is determined by variables B and E whereas the overtime, total vacant OR time and total master surgeon waiting time are respectively valued by O, V and W. Putting it all together yields the following CP formulation.

$$\min f(B, E, O, V, W, \varepsilon, \phi, \varphi) := (V \cdot c_V + W \cdot c_W + O \cdot c_O) \times \frac{\tau}{60} \tag{5.1a}$$

subject to :

$$\text{Sequential resource in } \phi(\sigma, \text{‘S’}), \tag{5.1b}$$

$$\text{Parallel resource in } \varphi(\sigma), \varepsilon(r), \tag{5.1c}$$

$$\text{Span}(\varphi(\sigma), s, \phi(\sigma, s)), \qquad \forall \sigma \in \Sigma \tag{5.1d}$$

$$\text{Count}(\sigma, \phi(\sigma, \text{‘P’}).\text{Begin}, 0, \text{‘>=’}, 1), \tag{5.1e}$$

$$\text{BeginAtEnd}(\phi(\sigma, \text{‘S’}), \phi(\sigma, \text{‘P’}), 0), \qquad \forall \sigma \in \Sigma \tag{5.1f}$$

$$\text{BeginAtEnd}(\phi(\sigma, \text{‘C’}), \phi(\sigma, \text{‘S’}), 0), \qquad \forall \sigma \in \Sigma \tag{5.1g}$$

$$O = \sum_{r \in \mathcal{R}} \max(0, \varepsilon(r).\text{Begin} - T), \tag{5.1h}$$

$$V = O + \rho T - \Psi, \tag{5.1i}$$

$$B = \min(\sigma, \phi(\sigma, \text{‘S’}).\text{Begin}), \tag{5.1j}$$

$$E = \max(\sigma, \phi(\sigma, \text{‘S’}).\text{End}), \tag{5.1k}$$

$$W = E - B - \varsigma, \tag{5.1l}$$

$$B, E, O, V, W \in \Theta. \tag{5.1m}$$

Since the goal is to minimize the total cost of the schedule, the objective function is computed as (5.1a) while the variables therein must satisfy several conditions. Constraint (5.1b) is a sequential resource on all the surgery activities of the different surgical procedures and therefore forces the solver to position the surgery tasks in different time periods. Constraint (5.1c) is a parallel resource on the surgical procedures and the last activities. It forces the completion of all the surgical procedures within the ORs by using activity level changes of 1 at the beginning/ending of each surgical procedure. It also computes the end time of each OR by using an activity level change of 1 at the beginning of each last activity $\varepsilon(r)$. This change keeps one OR used until the end of the schedule is reached. Constraint set (5.1d) creates a span for each surgical procedure's tasks. Constraint (5.1e) specifies that the work day of at least one the OR begins at time $t = 0$. Constraint (5.1f) exploits Proposition 1 meaning that the tasks of a surgical procedure are forced to be back to back in time as displayed in Figure 5.1. The interested reader will find our rationale in the last paragraphs of this section. The cleaning task is performed right after the surgery according to (5.1g). Constraint (5.1h) adds up each OR's potential overtime in accordance with the last activity's time flag thus computing the true overtime value. Constraint (5.1i) evaluates the vacant time for both ORs and constraints (5.1j)-(5.1l) compute the surgeon workload. The binding domain is defined in (5.1m).

Back to back rationale The back to back construction regroups the time variables of a surgical procedure under a unique entity, the surgical procedure span. That is to say when the solver decides a time position for a surgery σ, it needs not know the time positions of any other surgical procedures to establish the time frame of the surgical procedure which incorporates σ. Our understanding is that the propagation provided by global scheduling constraints is also more aggressive than comparison operators.

Indeed, think of an alternative construction ; the preparation time of a surgical procedure must be fixed according to the termination of its predecessor, if any. Although it is possible to create a model that finds such solutions directly by replacing constraints (5.1e)-(5.1f) with constraints (5.2)-(5.4), we argue that they are less intuitive. Case in point, the very statement of the alternative construction is conditional and is quite a testament to the difficulty of defining capturing constraints for a reality however basic it may seem.

$$\mathrm{Count}(\sigma, \phi(\sigma, \text{`P'}).\mathrm{Begin}, 0, \text{`='}, \rho), \tag{5.2}$$

$$\mathrm{Count}((\sigma_1, \sigma_2), \phi(\sigma, \text{`P'}).\mathrm{Begin}, \phi(\sigma, \text{`C'}).\mathrm{End}, \text{`='}, \mathcal{N} - \rho), \tag{5.3}$$

$$\mathrm{EndBeforeBegin}(\phi(\sigma, \text{`P'}), \phi(\sigma, \text{`S'})). \tag{5.4}$$

Two surgeons

In this case, all surgical procedures are scheduled back to back as both surgeons can work in parallel and the insertion of any delay could only increase the vacant time, overtime and total surgeon's waiting time. We recuperate some variables and activities from the previous model but modify some of them in order to know the OR associated with each task.

The idea is to duplicate the surgical procedure activities using an alternative format $\varphi_2(\sigma, r)$, $\forall(\sigma, r) \in \Sigma \times \mathcal{R}$. Once such a variable is selected, we know a surgeon is appointed to the task and it is then possible to compute the consequent cost. As such, the total surgeon waiting time W is distinguished by OR using variables $B(r)$ and $E(r), \forall r \in \mathcal{R}$. In the same vein, the end of the day is determined by variable $\varepsilon(r), \forall r \in \mathcal{R}$. The total overtime and vacant OR time remains easily tractable with single variables O and V. This gives rise to the following model.

$$\min f(B, E, O, V, W, \varepsilon, \phi, \varphi, \varphi_2) := (V \cdot c_V + W \cdot c_W + O \cdot c_O) \times \frac{\tau}{60} \tag{5.5a}$$

subject to :

$$\text{Sequential resource in } \varphi_2(\sigma, r), \qquad \forall r \in \mathcal{R} \tag{5.5b}$$

$$\text{Span}(\varphi(\sigma), s, \phi(\sigma, s)), \qquad \forall \sigma \in \Sigma \tag{5.5c}$$

$$\text{Alternative}(\varphi(\sigma), r, \varphi_2(\sigma, r)), \qquad \forall \sigma \in \Sigma \tag{5.5d}$$

$$\text{Count}(\sigma, \phi_2(\sigma, r).\text{Begin}, 0, \text{`='}, 1), \qquad \forall r \in \mathcal{R} \tag{5.5e}$$

$$\text{BeginAtEnd}(\phi(\sigma, \text{`S'}), \phi(\sigma, \text{`P'}), 0), \qquad \forall \sigma \in \Sigma \tag{5.5f}$$

$$\text{BeginAtEnd}(\phi(\sigma, \text{`C'}), \phi(\sigma, \text{`S'}), 0), \qquad \forall \sigma \in \Sigma \tag{5.5g}$$

$$\varepsilon(r) = \max_{\sigma \in \Sigma}(\varphi_2(\sigma, r).\text{End}), \qquad \forall r \in \mathcal{R} \tag{5.5h}$$

$$O = \sum_{r \in \mathcal{R}} \max(0, \varepsilon(r) - T), \tag{5.5i}$$

$$V = O + \rho T - \Psi, \tag{5.5j}$$

$$B(r) = \min_{\sigma \in \Sigma}((1 - \varphi_2(\sigma, r).\text{Present}) \cdot T + \phi(\sigma, \text{`S'}).\text{Begin}), \qquad \forall r \in \mathcal{R} \tag{5.5k}$$

$$E(r) = \max_{\sigma \in \Sigma}(\varphi_2(\sigma, r).\text{End} - \pi(\sigma, \text{`C'})), \qquad \forall r \in \mathcal{R} \tag{5.5l}$$

$$W = \sum_{r \in \mathcal{R}}(E(r) - B(r)) - \varsigma, \tag{5.5m}$$

$$O, V, W \in \Theta, \tag{5.5n}$$

$$B(r), E(r), \varepsilon(r) \in \Theta, \qquad \forall r \in \mathcal{R} \tag{5.5o}$$

Constraint set (5.5b) are sequential resources on all the surgical procedures $\varphi_2(\sigma, r)$ of OR r. Constraint set (5.5d) establishes on which OR a surgical procedure is executed ; these constraints work only if the surgical procedures $\varphi_2(\sigma, r)$ are defined as optional activities because they must be present in a unique OR r for each surgical procedure σ. Constraint (5.5e) specifies that the work day of each OR begins at time $t = 0$. Constraint set (5.5h) computes the end of the day in OR r. Constraints (5.5i) and (5.5j) evaluate respectively the overtime and the vacant time for both ORs. Since the surgical procedures are now executed back to back, there cannot be overtime and vacant time in the same OR. Constraints (5.5k)–(5.5m) compute the surgeon workload. Since the value of $\varphi_2(\sigma, r).\text{Begin} = 0$ if surgery σ is not present in OR r, we have to increment the time of beginning by a factor T in (5.5k). The binding domain is defined in (5.5n)–(5.5o). Finally, constraints (5.5c), (5.5f) and (5.5g) work the same as in the 1-to-2 model.

5.2.4 Stochastic case

Let us introduce the optimization model as a paradigm rather than a model. Indeed, we capitalize on the promising results of the deterministic model to build a stochastic algorithm. The idea is to extract enough information, a la Monte Carlo, about the uncertainty of the surgeries' time requirement to propose a good schedule. In order to measure the quality of a schedule, it is important to consider how well it copes with the instability of the actual surgical procedures times.

In this matter, online realizations of surgical procedure times are likely to perturb any given schedule. Then again, observe that neither the ordering nor the room assignment are influenced by this perturbation. A *timeless* schedule allows one to experience the time period component at the end of the day in light of observed process times. It is then possible to compute the *real* cost of this timeless schedule. The exercise of computing the cost of a timeless schedule can be done with a straightforward brute-force approach that has almost no computational cost. Let us call this exercise *pricing*.

In the spirit of the SAA method (Kleywegt et al., 2002), the stochastic algorithm works by using two sets of scenarios : the *profile set* $\Omega_1 := \{\omega_{1r}\}_{r\in\{1,\dots,n_1\}}$ and the *sample set* $\Omega_2 := \{\omega_{2i}\}_{i\in\{1,\dots,n_2\}}$. The first stage of the stochastic algorithm constructs a set of timeless schedules using the profile set Ω_1. For each scenario $\omega_{1r} \in \Omega_1$, a schedule is established by an oracle whereby the time period is neglected providing a timeless schedule r. The latter is stored in R_{Ω_1}. The second stage prices every timeless schedule against every scenario of the sample set Ω_2.

Opposite the sample set which serves to capture the probabilistic nature of the process times, we have the profile set whose every element produces a timeless schedule which is tested against every possibility contained in the sample set. The retained schedule therefore uses one of the timeless schedules along with time periods determined during the stochastic optimization, and its selection highlights the fact that it tackles randomness better than others.

One would like to select a sample size n_2 sufficiently large such that, by the law of large numbers, the only culprit of the algorithm lies in the possibility that the profile set is unable to produce the *best* schedule. In other words, when the sample set provides converging statistics, a random profile set can identify the best stochastic schedule with probability 1 as $n_1 \to \infty$

(assuming a capable oracle of course).

The sets are built using the empirical data and the principle of indifference for each task's process time such that realization values are randomly selected using a uniform distribution. Of course, model fitting would be a welcome addition that would increase the stochastic model's quality but requires an additional analysis of the limited empirical data for which we have no feedback.

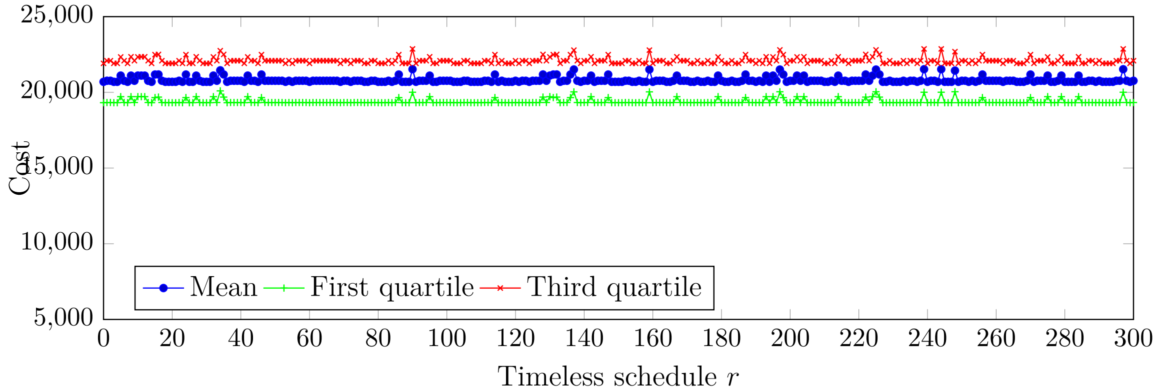
Recall that the pricing exercise is trivial. The bottleneck of the algorithm is therefore the resolution of the oracle for the n_1 timeless schedules. We ultimately obtain a cross matrix A which compares all scenarios and schedules to each other.

1- create a **pool** R_{Ω_1} of timeless schedules :
for all $\omega_1 \in \Omega_1$ **do**
  solve the oracle
  store the timeless schedule r in R_{Ω_1}
end for
2- **price** all timeless schedules :
for all $r \in R_{\Omega_1}$, $\omega_2 \in \Omega_2$ **do**
  price the timeless schedule r on the scenario ω_2
  store this price in the matrix A
end for
3- **compute** the function g
4- **select** the timeless schedule r which reaches the maximum of g

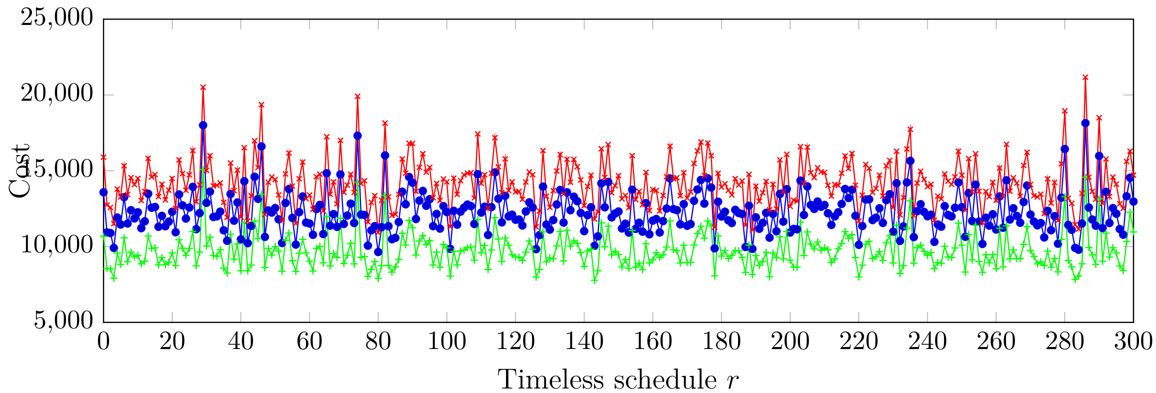
Algorithm 3 Sample Average Approximation Method

Different statistics can be extracted from the matrix A. Let m_r be the mean, Q_r^2 the median, Q_r^1 the first quartile and Q_r^3 the third quartile of the costs of the schedule r. Further define M and v as the mean and standard deviation of m_r. For each schedule r, we additionally compute the percentage p_r of scenarios which have a cost lower than $M + v$. The mean represents the expected cost of a schedule on the sample set Ω_2. The other statistics give an idea of the dispersion of these costs.

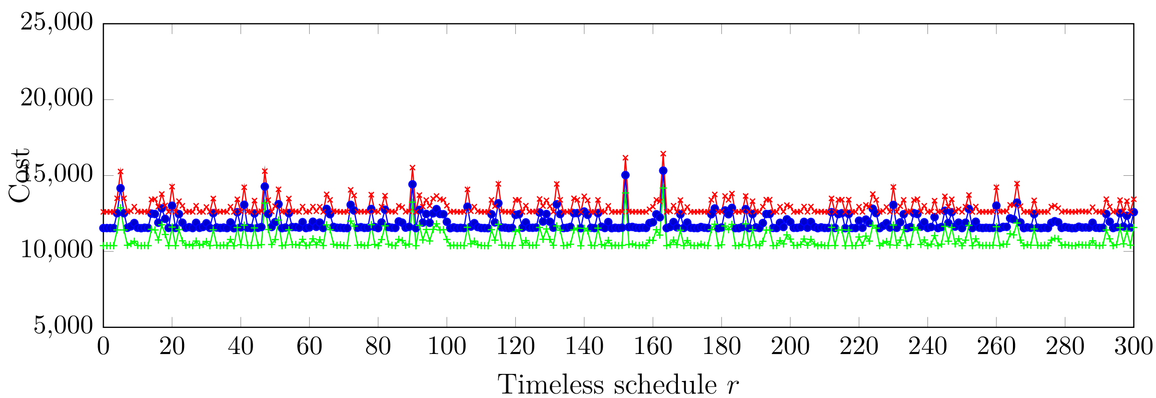
Figure 5.2 shows the mean, the first quartile, and the third quartile for the first 300 schedules obtained during the stochastic resolution of instance 7. One could decide to compute any other relevant statistics for their problem. The best schedule in terms of expected cost is not necessarily the most stable. Indeed, two schedules can have approximately the same expected cost but a totally different dispersion. As such, schedules are obtained using the profile set whereas their robustness is tested against the sample set. Let the idea of a quality schedule be encapsulated by a robustness measure g which computes the best 5% schedules for different



(a) One OR with one surgeon



(b) Two ORs with one surgeon



(c) Two ORs with two surgeons

Figure 5.2 – Distribution of m_r, Q_r^1 and Q_r^3 (instance 7)

measures : m_r, Q_r^2, Q_r^3 and p_r. If a schedule is in this ranking, it obtains 1 point (2 points for the mean) and 0 otherwise. Finally, the schedule with the best score g is retained whereas the means are compared in case of a tiebreaker. This measure gives an important weight to the expected cost of schedule, but it also takes into account the cost dispersion in order to ensure a good upper bound on the latter.

Deeper measures

The proposed statistics can be seen as elements of a dashboard for the retained schedule. Observe that these statistics are not taken into account in the optimization model because they might make the problem infeasible or take too much time to solve. They do however give extra criteria to evaluate the different schedules. Our algorithm allows their computation for all combinations of timeless schedule $r \in R_{\Omega_1}$ and scenarios $\omega_2 \in \Omega_2$. The truth is any other statistic could be incorporated in the same way. While it would require some analysis to determine an appropriate weight, we shall let this open to debate for future research. Algorithm 3 ultimately allows the performance measure g to be as complex as one wishes allowing it to reflect the manager's experience and risk tolerance.

Although the statistics presented are restricted to cost metrics, an avid manager could incorporate other practical performance metrics. One can indeed define relevant statistics according to hospital goals. Let us give two examples : first, a critical time period span T^C that goes hand in hand with an overload or underload flag for the number of surgeries, and second, discriminative arrival times for patients to reduce the overall patient waiting time.

A critical time period span $T^C > T$ could be set and we could measure the proportion of scenarios in which one schedule r exceeds this length. It is also easy to see if the number of surgical procedures scheduled is acceptable. If most of the scenarios have the last procedure finish a long time before T, the manager would know that another one could fit in the day. The opposite can be said if the last preparation begins after T in most of the scenarios.

As it stands today, the underlying assumption of any given operating room schedule is that all patients are present on time to undergo their surgery. In order to validate this assumption, it is not unusual for hospitals to ask *all* patients to present themselves at some arbitrary time. While this guideline changes nothing for the first patient of the day, it is an entirely different story for the last one. It turns out, we could easily set an arrival time for every patient which would reduce their waiting time compared to an environment where every patient would arrive at the beginning of the day by simply looking at the earliest start time for individual patients in the Ω_2 scenarios.

In order to ascertain the aforementioned assumption, we can use the time distribution to compute a maximal arrival time for each patient with a given certainty rate of $p\%$. Observe that a certainty rate of 100% amounts to the traditional arbitrary time rule, while allowing a 1% risk tolerance is likely to produce considerable reduction in patient waiting time.

5.3 Results

We are ready to present the different results obtained using the CP models. We used the modeling software `AIMMS 3.14` and the incorporated solver `CP Optimizer 12.6` on a computer running an `Intel` i7-4510U@2.0Ghz along with 8Gb of RAM. As part of the `IBM ILOG CPLEX Optimization Studio`, default solver parameters and propagating methods turned out to be plenty powerful. Section 5.3.1 brings some details about the data that we used to run our models. Sections 5.3.2–5.3.3 follow the same organization as before and respectively address the deterministic and stochastic models. We further venture several insights and openings about generalizing our work by taking into account practical concerns in each of these.

5.3.1 Problem data

A total of eleven surgery types are used to build eleven instances of $\mathcal{N}$ surgeries (ranging from 4 to 11) over a regular day time span of T (4, 8 and 12) hours. Table 5.1 summarizes their specifications where the last two columns indicate the *workload* and *surgery load* ratios of the instance with respect to the regular time operating hours. Both ratios are expressed as a percentage of T whereby the first measures the quantity of work while the second the quantity of work for the surgeon. The reader might be intrigued about the seemingly out of order eleventh instance. The latter is a custom instance whose purpose is to show the benefit of a 2-to-2 scheme in the stochastic environment whereas the first ten instances are part of the AIMMS-MOPTA 2013 Challenge (2013). Finally, the other pertinent parameters $c_V :=$ $1209.60, c_W := 1048.80$ and $c_O := 806.40$ are the three main costs respectively associated with the vacant time, surgeon waiting time, and overtime. A time discretization of $\tau := 5$ is retained.

5.3.2 Deterministic cases

We first present the 2-to-1 results and then move on to the extreme cases, that is those of the 1-to-1 and 2-to-2 models. The comparison that follows tries to put these results in perspective with respect to the operating theatre problem.

Table 5.1 – Description of the 11 instances

Instance	T (hours)	$\mathcal{N}$	Sum of processing times (% of T) all tasks	surgery tasks
1	4	4	102%	41%
2	4	5	125%	51%
3	4	5	152%	63%
4	8	6	116%	51%
5	8	7	129%	50%
6	8	10	186%	67%
7	8	11	262%	97%
8	12	7	103%	49%
9	12	10	170%	63%
10	12	11	216%	92%
11	8	10	238%	125%

One surgeon and Two ORs

In order to evaluate the quality of our CP model, we use two constructive procedures highly popular for this type of problems (see Harper, 2002) and (Huschka et al., 2007) ; shortest processing time (SPT) and longest processing time (LPT). Only the surgery length was taken into account for the heuristics and the next surgery is placed in the first free OR. The deterministic results for one surgeon and two ORs are summarized in Table 5.2.

Table 5.2 – Deterministic results (1 surgeon and 2 ORs)

	Heuristics		Optimization model		Optimality gaps (%)		
Instance	SPT	LPT	CP	Time (s)	SPT	LPT	CP (1 s)
1	4798.2	5235.2	**4710.8**	0.00	1.9	11.1	0.0
2	4402.0	3790.2	**3615.4**	0.02	21.8	4.8	0.0
3	2903.4	2291.6	**2204.2**	0.02	31.7	4.0	0.0
4	8501.0	8413.6	**8238.8**	0.19	3.2	2.1	0.0
5	7863.0	7688.2	**7163.8**	0.61	9.8	7.3	0.0
6	8368.0	8978.8	**2493.8**	2.72	235.6	260.0	7.0
7	10370.4	10221.8	**6210.2**	25.89	67.0	64.6	0.0
8	15147.4	14623.0	**14273.4**	0.38	6.1	2.4	0.0
9	6943.0	5719.4	**5195.0**	2.02	33.6	10.1	0.0
10	7487.4	8891.6	**4254.6**	29.38	76.0	109.0	10.1
11	8111.4	10792.6	**6417.8**	11.77	26.4	68.2	12.0

The computation times are omitted for the heuristics because they are virtually null. The results speak for themselves, both heuristics are instantly discarded with respect to the power of CP which provides optimal solutions (in **bold**) within thirty seconds for all instances. As a matter of fact, the last three columns contain the optimality gaps for both heuristics as well as the CP model using a one second cut-off. Reaching up to 260% on the resolution of instance 6 with the LPT method, these gaps pale in comparison to the mere 7.0% offered by the cut-off option. Further observe that the heuristics perform unevenly against one another and are systematically outperformed either way. Finally, one can observe that the heuristics behave particularly poorly on instances 6, 7, 9, 10 and 11 as supported by the longer running times observed with the CP model. As it stands, it seems that the workload ratio is a good difficulty indicator.

Extreme cases

For some of the instances evaluated, the total procedure time is much smaller than twice the regular day time span (time allowed if two ORs are open) which leads to large vacant times. This motivates the comparison of the previous solution to one where there is only one OR assigned to a surgeon. The result of this decision would be a decrease in vacant time at the cost of an increase in the surgeon's idle time. The manager would benefit of this situation if the vacant costs are greater than the surgeon's idle costs. It would also give the possibility of allowing another surgeon to perform procedures in the second OR or reducing the fixed costs of opening the other OR. We have also considered the opposite case where two surgeons worked in two ORs. The results obtained from the CP formulation are summarized in Table 5.3.

Given a unique surgeon, these results show that opening only one OR can be cheaper than opening both. Part of the explanation resides in the relatively low workload such as instances 1, 2, 4 and 8 present. We indeed say part since instance 5 also has a low workload yet two ORs is still preferable. At the same time, it can be very expensive in overtime and in waiting time for the surgeon, if there are not enough ORs opened to realize all the surgeries. In this deterministic environment, the last case with 2 surgeons assigned to 2 ORs is never the best option for these instances. This ultimately means that a good strategic and tactical planning is of the essence.

5.3.3 Stochastic cases

This section is organized similarly to the deterministic cases although we first take the time to discuss parameter selection relating to the SAA scheme, that is the oracle used to fetch

Table 5.3 – Deterministic comparison of 1 vs 2 surgeons and 1 vs 2 ORs

	One Surgeon		Two Surgeons
Instance	One OR	Two ORs	Two ORs
1	**1687.4**	4710.8	5147.8
2	**2709.0**	3615.4	4576.8
3	4349.0	**2204.2**	3777.4
4	**4920.8**	8238.8	10773.4
5	7676.8	**7163.8**	10922.0
6	12858.6	**2493.8**	6776.4
7	20709.0	**6210.2**	11461.0
8	**5580.0**	14273.4	17769.4
9	16905.4	**5195.0**	11487.8
10	22974.4	**4254.6**	10500.8
11	16339.8	**6417.8**	8577.6

timeless schedules as well as the size of the sample and profile sets.

Parameter selection

While the parameter selection discussion revolves around instance 7, an instance which arguably seems to benefit significantly from the stochastic analysis as supported by the very high VSS and EVPI result values (see Table 5.5), we invite the reader to understand the assertions more generally as consistent behavior is observed across the test bed.

Oracle performance. Recall the one second cut-off timer of the CP solver discussed with the deterministic computing times in Table 5.2. Figure 5.3 compares the evolution of the minimal average cost achieved within a given computing time, that is, a number of evaluated profiles, for three deterministic oracles : LPT, SPT and CP (1 s). Obviously, for each oracle the expected cost of the minimum solution decreases as the computation time increases ; however the CP solver levels off faster than both heuristics. After one hour of computation, the LPT heuristic has priced approximately 3,300 profiles, the SPT 5,000, and the CP 1,100.

Furthermore, Table 5.4 exhibits how the best retained solution is always given by the CP solver even if the two heuristics are able to price more timeless schedules. The CP solver is thus chosen as the deterministic oracle of Algorithm 3.

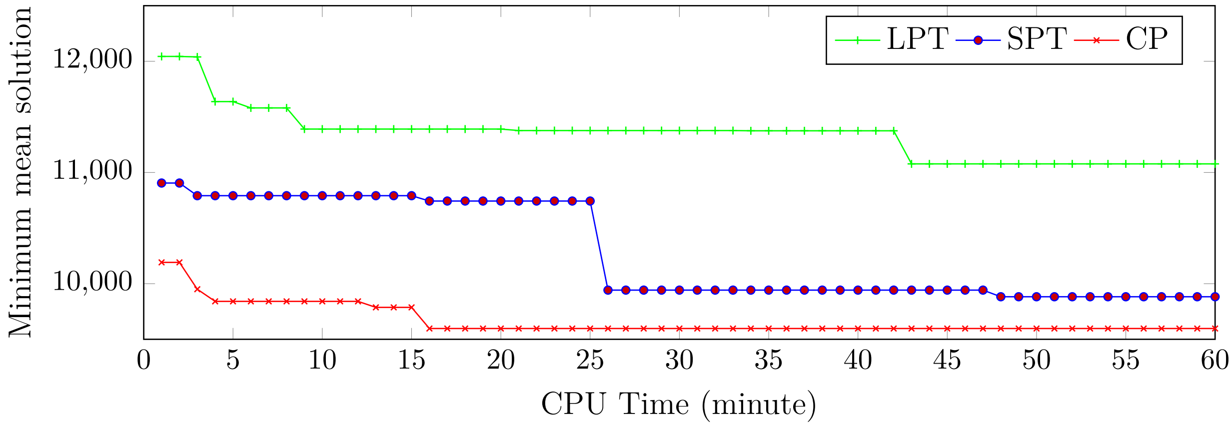


Figure 5.3 – Best mean evolution opposite computation time in minutes (instance 7)

Table 5.4 – Retained solution gaps (in %) for the heuristic oracles

Instance	1	2	3	4	5	6	7	8	9	10	11
SPT	2.6	4.1	0.0	0.0	1.4	15.3	3.0	1.6	12.6	1.4	4.9
LPT	1.2	0.0	0.4	0.4	0.3	3.8	15.4	0.2	0.1	18.9	25.0

Sample and profile sets sizes. Figure 5.4 depicts the evolution of the perfect information (see Table 5.5 for a definition). A sample size $n_2 = 1,500$ secures enough stability for the resulting solutions to be meaningful whereas the profile set size n_1 remains variable such that one hour of computing time is granted to each instance.

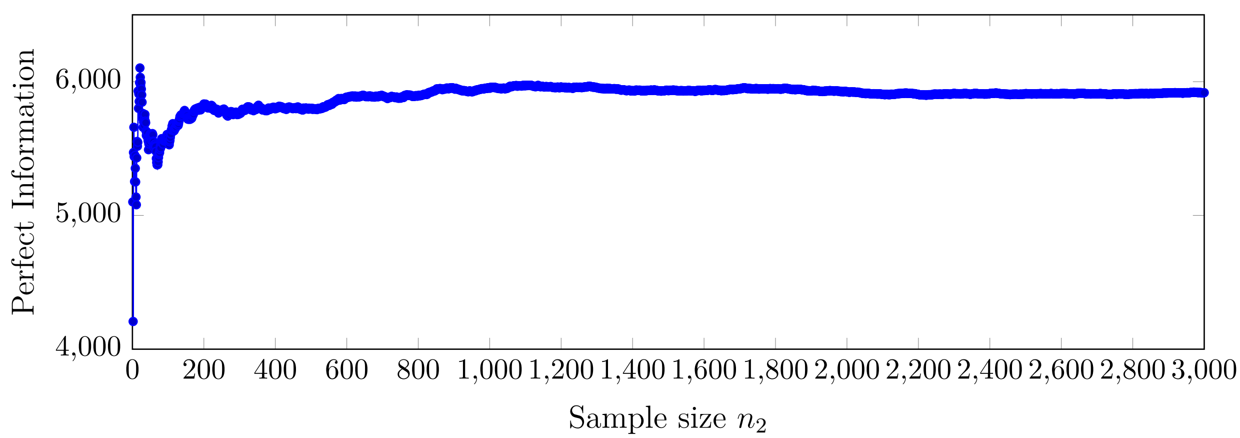


Figure 5.4 – Perfect information evolution opposite sample size n_2 (instance 7)

One surgeon and two ORs

Let us call the 3-σ rule, a heuristic that aims to incorporate the variability of the process times for the different surgeries. Define $\mu(\sigma)$ as the average process time of the surgery σ plus three times its standard deviation. The latter can be thought of as an upper bound for the length of said surgery. Indeed, under a normal distribution, over 99% of the observed values would be smaller than $\mu(\sigma)$. The surgeries with the largest $\mu(\sigma)$ are realized first. Therefore, the surgeon has a high probability to perform the longest surgeries first. Algorithm 4 builds the schedule with the 3-σ rule.

H <- surgeries sorted by decreasing $\mu(\sigma)$
while H is not empty **do**
  remove first surgery from H
  put this surgery in the first free operating room
end while

Algorithm 4 Stochastic 3-σ Rule of Thumb

Table 5.5 presents the expected value of each solution. The 3-σ rule solution is computed by Algorithm 4. The deterministic solution is the expected cost of the schedule obtained by the CP model (5.1) on the deterministic case. The minimum solution is the one which reaches the minimum expected cost. The retained solution is computed by Algorithm 3. The perfect solution is the mean of the minimum cost obtained by the CP model (5.1) on each scenario ω_2. Since it is impossible to have better results, it can be seen as the lower bound of the expected value. Finally, we look at two other measures to see the contribution of the stochastic approach. The value of the stochastic solution (VSS) measures the gain of this approach, i.e., the difference between the expected costs of the deterministic and the retained solutions. Observe that although highly unlikely a negative VSS value is possible. Indeed, since the retained solution is based on the performance measure g, its expected cost can be higher than the deterministic one. The expected value of perfect information (EVPI) shows the maximum that one would be able to gain with a better procedure, i.e., the difference between the expected cost of the retained and the perfect solution. For the sake of thoroughness, recall that these stochastic measures are dependent on the profile and sample sets. Consequently, Algorithm 3 might be improved by getting a better estimation of the future or using a larger profile set.

As expected, the stochastic environment penalizes significantly non adaptive designs such as the deterministic solution or the 3-σ heuristic. Furthermore, the retained and the minimum solutions are always the same except for the instances 1 and 5 (in **bold**). It is the perfect

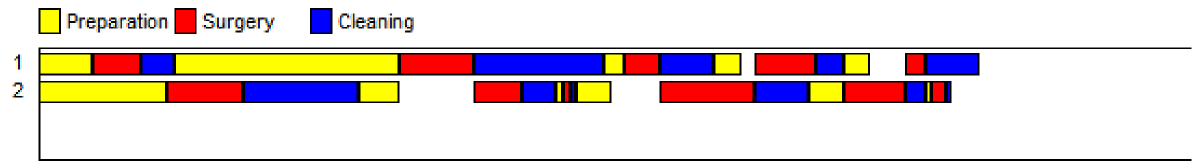
Table 5.5 – Stochastic results (one surgeon and two ORs)

	Expected cost (m_r) of the solution						
Instance	3-σ rule	Deterministic	Minimum	Retained	Perfect	VSS	EVPI
1	5389.2	4923.0	**4864.1**	4881.7	4786.2	58.9	77.9
2	4042.7	3870.8	3870.8	3870.8	3706.6	0.0	164.3
3	3239.6	3074.9	3061.3	3061.3	2499.1	13.6	562.2
4	8788.9	8716.4	8533.9	8533.9	8133.9	182.5	400.0
5	8100.2	7815.4	**7771.8**	7777.0	7317.6	38.4	459.4
6	13000.6	5698.1	5546.8	5546.8	3224.0	151.3	2322.8
7	13507.1	10872.6	9596.8	9596.8	5951.2	1275.8	3645.6
8	15129.4	15048.8	14840.7	14840.7	14335.7	208.0	505.0
9	8494.0	8032.4	7706.4	7706.4	5403.0	326.0	2303.4
10	14221.8	9965.8	9894.6	9894.6	3922.7	71.2	5972.0
11	14997.0	9517.4	9260.5	9260.5	5015.0	256.9	4245.5

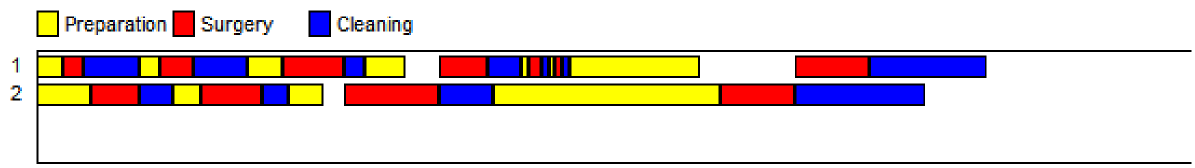
opportunity to underscore the reason we did not only choose the expected cost to measure the quality of a solution. Other statistics such as the third quartile can improve the stability of the solution cost. Indeed, for the instance 5, the minimum solution does not reach a good enough percentage on this statistic. This comes at the expense of a small fee, i.e., 5.2 = 7777.0 - 7771.8. This fee still falls within the realms of expected values and can be seen very much like an insurance fee for peace of mind. In other words, our algorithm is sufficiently flexible to permit the manager to find an equilibrium in accordance with his risk aversion.

The VSS proves that the stochastic approach is worth it. The gains are important for all instances. Furthermore, the EVPI shows that it is possible to gain even more in terms of expected cost. If the uniform distribution does not fit well the observed data, another distribution could improve a lot of the results and decrease the EVPI while increasing the VSS. However, the EPVI gives the gap with the oracle (and optimal) solution, this solution is impossible to reach because it adapts the schedule perfectly for each scenario.

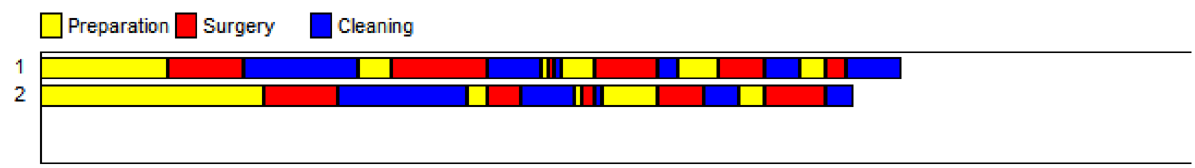
It can be also noted that the instances 6, 7, 9, 10, and 11 remain difficult to optimize in the stochastic case as supported by the very high EVPI values. Part of the explication resides in the fact that these instances have a lot of surgeries but the most prominent explanation lies in the very construction of the retained schedule. The stochastic algorithm faces two conflicting objectives in that one aims to minimize the expected cost whereas the other aims to select a schedule which behaves well against several scenarios. Indeed, since a great deal of effort is deployed to minimize the objective during the CP resolution, most of the tasks are realized one right after the other. Furthermore, the surgeon performs surgeries over two ORs, consequently most of the surgery tasks are on the critical path. Any perturbations in the process times is thus likely to strongly impact the schedule as shown in Figure 5.5. Indeed, the latter proposes different schedules computed on the same scenario from 3 timeless schedules : the deterministic, the stochastic, and the perfect information (which corresponds to the optimal schedule). While the optimal schedule copes perfectly with the uncertainty, the deterministic schedule offers a planning with a lot of idle time. Comparing to the deterministic, the stochastic schedule decreases a little the 3 costs : the surgeon waiting time, the vacant time, and the overtime. Finally, the next section continues to analyze the effect of opening two ORs with one surgeon in a stochastic and more realistic case.



(a) Deterministic schedule



(b) Retained schedule



(c) Optimal schedule

Figure 5.5 – Gantt charts on a random scenario (instance 7)

Extreme cases

Table 5.6 compares the expected cost of the retained solution over the sample scenarios for all three cases : 1-to-1, 1-to-2 and 2-to-2. The relative change column measures the difference with their deterministic *vis-a-vis* of Table 5.3 using the latter as reference.

It seems that when there are a lot of surgeries to schedule (instances 3, 6, 7, 9, 10, and 11), the relative change due to the variability of the process times is very important for the 1-to-

Table 5.6 – Stochastic comparison of 1 vs 2 surgeons and 1 vs 2 ORs

	One Surgeon				Two Surgeons	
	One OR		Two ORs		Two ORs	
Instance	Retained	Relative change (%)	Retained	Relative change (%)	Retained	Relative change (%)
1	**1870.0**	10.8	4864.1	3.3	5252.9	2.0
2	**2675.5**	-1.2	3870.8	7.1	4635.6	1.3
3	4171.0	-4.1	**3061.3**	38.9	3921.7	3.8
4	**5045.2**	2.5	8533.9	3.6	10839.2	0.6
5	**7241.5**	-5.7	7777.0	8.6	11107.0	1.7
6	12402.4	-3.5	**5546.8**	122.4	7598.8	12.1
7	20749.3	0.2	**9596.8**	54.5	11558.5	0.9
8	**6285.7**	12.6	14840.7	4.0	17833.7	0.4
9	16790.7	-0.7	**7706.4**	48.3	11811.2	2.8
10	23376.5	1.8	**9894.6**	132.6	11656.3	11.0
11	16840.1	3.1	9260.5	44.3	**9217.3**	7.5

2 configuration. In fact, one of the most worthwhile observation concerns how the relative changes are orders of magnitude higher for this configuration compared to the extreme cases where these hover around zero.

In both extreme cases, as the surgeon and the OR become indistinguishable the resource constraints binding them together becomes redundant. When an OR is assigned to a specific surgeon, the variability of the process times that can be observed on the different surgeries is absorbed by the compensation these operations procure to one another. The total cost is eventually only affected by the first and last surgeries. Taking a look at Figure 5.2, it should then come as no surprise that the 1-to-2 configuration is the most unstable.

Compared to deterministic results, another notable fact is that instances 5 and 11 perform better on different configurations, that is respectively 1-to-1 and 2-to-2. In fact, while the 1-to-2 configuration remains a perfectly well suited option as testified by five favorable instances, the deterministic configuration conclusion regarding 1-to-2 is mitigated in the stochastic environment. Indeed, as the gap between the 1-to-2 configuration and the extreme counterparts narrows, the most suitable configuration option may or may not change. Ultimately, this means that in order for the management of operations to be effective, one needs to consider the stochastic aspect explicitly.

5.4 Conclusion

We believe the OR management problem is a very important issue in today's world and it stands to reason even more papers will address this problem in the near future. Our integrated approach to lift the uncertainty dimension relies on our capability to solve a deterministic version in an efficient and reliable manner. Among the different methods addressed, we have shown that constraint programming outperforms the SPT and LPT heuristics both in the deterministic and stochastic environments.

From a management perspective, we can extract two types of decisions from our analysis. The first is a tactical decision and corresponds to operating room management guidelines. Depending on the variability of the surgeries to perform, a policy can be devised to establish the trade-off between ORs and surgeon costs. Three options are available : one OR and one surgeon, two ORs and one surgeon and finally two ORs and two surgeons. Although the first and third options have the advantage of being more stable in a stochastic, read realistic environment, the second does best often nonetheless.

The second is an operational decision and depends on the risk aversion of the manager. Indeed, the tactical policy determined which resources the retained schedule may utilize. Its cost can be seen as a twofold measure : an expected cost plus a security loading which allows said schedule to behave more consistently. In the end, the size of this security loading is left at the discretion of the manager.

As an open question, we raise the possibility of a more integrated model where the length of regular time is also taken as a variable rather than an input. This might lead to cost reduction for the idle OR after the last procedure and could provide a good approximation for the cost of actually opening an OR room in a given hospital.

CHAPITRE 6 ARTICLE 3 : ONLINE STOCHASTIC OPTIMIZATION OF RADIOTHERAPY PATIENT SCHEDULING

A. Legrain, MA. Fortin, N. Lahrichi et LM. Rousseau ont écrit cet article et l'ont publié en 2014 dans *Health Care Management Science*.

6.1 Introduction

The Canadian Cancer Society (Canadian Cancer Society's Advisory Committee on Cancer Statistics, 2013) reports that "on average, one person learns that he or she has cancer every 3 minutes in Canada. Every 7 minutes, one person in Canada dies from cancer." Therefore, it is crucial to ensure that all Canadians have good access to cancer care. In 2011, Quebec authorities inaugurated two new cancer treatment facilities in the Montreal region. We collaborate with one of them : the Centre Intégré de Cancérologie de Laval (CICL).
The services provided by the center include chemotherapy and radiotherapy ; the latter uses radiation to destroy malignant cells to prevent them from multiplying. Radiation treatments are administered with specialized equipment such as linear accelerators (linacs). Each patient who arrives at the CICL for radiotherapy must leave the center with an appointment for the linacs. Furthermore, the patients have different specificities that are not known in advance : their day of arrival, their priority, and their treatment duration. They also share the same resources at the CICL. This complicates the scheduling of the linacs and the choice of the first day of treatment.

The objective of this paper is to solve the radiotherapy patient scheduling problem in an online fashion. We aim to ensure that the patients receive an initial treatment within a reasonable time and that there is an optimal use of the resources, which are mainly linacs.
We present an innovative algorithm to solve the online scheduling problem. We implement a hybrid method combining stochastic optimization and online optimization. Our work builds upon the online stochastic algorithm presented by Legrain and Jaillet (2013). Our main contributions are the adaptation of this algorithm and its application to real instances. The modifications allow us to model more complex and realistic problems. After various computational tests to set the parameters of the online stochastic algorithm, we solve two real instances. The results outperform the solutions used by the CICL. The improvements are at least 300% as measured by our objective function.

In the next section, we present the radiotherapy scheduling problem and a literature review that places our contribution in the context of previous work on radiotherapy scheduling and online problems. In Section 6.3, we build an offline model and then an online procedure based on this model. In Section 6.4, we test our algorithm in a theoretical context and on two real instances. We conclude with some final remarks and ideas for future developments.

6.2 Problem statement and related work

We address operational management issues in this paper. The CICL administrators face several performance-related challenges. The provincial management closely monitors the access to specialized services, following up on individual patients on the waiting list. In Quebec, the waiting-time target for radiation oncology is set to four weeks (for 90% of the patients) and it is calculated as the time elapsed between the day when the patient is ready for treatment and the first day of treatment (Ministère de la santé et des services sociaux, 2010). Many studies have shown that a delay in starting radiotherapy has a negative effect on the patient's clinical condition(Chen et al., 2008). The impact of such a delay varies greatly depending on the nature and the severity of the cancer and the progression of the disease. The Ministry of Health and Social Services (MHSS) advises meeting certain deadlines depending on the patient's condition.

A radiation oncology department serves both curative and palliative patients, and they have different priorities. Seventy percent of the patients have a curative profile. After a complex treatment-planning process involving several health-care professionals, the treatments are delivered 5 days a week, for a total of 5 to 44 treatments. The remaining 30% of the patients are palliative and need rapid access to care (in less than 3 days) to be relieved of serious pain.

Before beginning treatment, the patient must undergo a CT scan and a specialized team (radiation oncologist, dosimetrist, physicist, and technologist) must perform a sequence of complex tasks. The efficiency of CICL operations depends on the managers' ability to use their resources to their full potential to ensure that the patients are treated on time. Since the linacs are in great demand in a cancer treatment facility, we focus on scheduling appointments and booking slots on the accelerator. The CICL has decided to divide each linac schedule into time slots of twenty minutes, and assign an appointment to each slot. The center has a policy of treating curative patients at the same time every day. However, palliative patients can be treated at different times. Therefore, the problem involves assigning a first day of treatment and a slot on the accelerator for each patient.

The treatment requires several time slots, so the clerk must make several appointments for the patient while ensuring that the clinical and operational constraints are satisfied. He must also leave free slots in the schedule to allow for the possible arrival of more urgent patients.

The daily management of unforeseen events is important given the cost of cancer treatment. In Quebec this cost was estimated in 2008–2009 at \$4400, i.e., \$454 million for 103,000 patients (Boucher and Beauregard, 2010). This includes only the operational costs : workforce costs, supplies, and equipment maintenance. The random patient inflow must consequently be managed to minimize the impact on operations. To control the impact of unforeseen events, the appointment scheduling must include a stochastic aspect. Some patients undergo surgery or chemotherapy before starting radiotherapy treatment. The CICL can access information for about 80% of these patients four weeks in advance, and the stochastic optimization can take this information into account.

The appointment optimization literature is wide ; the reader is referred to (Cayirli and Veral, 2003; Gupta and Denton, 2008; Batun and Begen, 2013) which are three good reviews in this field. Researchers have drawn attention to many applications. In (Begen and Queyranne, 2011), the single-server appointment scheduling problem where a doctor for example must give a time of appointment to a sequence of patients is tackled. Santibáñez et al. (2007) present the appointment scheduling of operating rooms. Patrick et al. (2008b) study the appointment scheduling of patients with different priorities on a single server (for example a scanner). However, the optimization of appointments for radiation therapy is a relatively young field with a limited literature. Furthermore, CICL policy requires that patients leave the center with their appointment, so an online algorithm is necessary. This review aims to show how the radiotherapy patient scheduling problem can take advantage of online optimization techniques which have a mature literature.

6.2.1 Radiotherapy booking

Several simple strategies for the scheduling of a patient's radiotherapy are described in (Petrovic et al., 2006). In particular, the "as soon as possible" (ASAP) strategy gives satisfactory results for this problem. The patients are sorted in ascending order of deadline and given the earliest possible appointments. Other approaches are proposed in (Petrovic and Leite-Rocha, 2008). Several ideas are tested, e.g., postponing the assignment of appointments or imposing limits on the use of resources. These ideas can easily be used by a cancer treatment facility.

Kapamara et al. (2006) provide a good review of exact methods and metaheuristics for the radiotherapy scheduling problem. Conforti et al. (2008, 2009) and Jacquemin et al. (2010) present techniques based on linear programming. However, these methods do not take into account the arrival distribution of the patients or future events. One contribution of our paper is a novel method that takes these two features into account.

To the best of our knowledge, stochastic optimization has been used only once in radiation oncology. Sauré et al. (2012) introduce the radiotherapy patient scheduling problem and solve it using techniques presented in Patrick et al. (2008b). They take into account the different patient priorities and use a dynamic approach to deal with uncertain arrivals. They develop a method based on a Markov decision process and achieve very good results on randomly generated instances based on statistics from the British Columbia Cancer Agency. However, the Markov decision process computes an optimal policy (105 min of computing time) in advance and becomes effective when the process reaches a steady state. The CICL has advance information for some patients, and it is difficult to integrate such information into this method. Indeed, it is necessary to recalculate the optimal strategy with this information and to wait for the process to return to a steady state : information about the future has already been revealed at this point, and the recalculation remains computationally demanding. Erdogan and Denton (2013) present a model for booking daily appointments in an online fashion ; the instances are small (20 patients) but complex. They use stochastic optimization to deal with the uncertainties. Since the patient arrival distribution of the CICL changes dynamically for each patient arrival, stochastic and online optimization techniques are more appropriate.

6.2.2 Online optimization

Online optimization techniques have been developed to make a decision quickly when an event occurs while maintaining a good solution. For example, advertisement placement by search engines (the Adwords problem) is a well studied problem. For each user search, the engine displays advertisements related to the keywords of the search ; the goal is to maximize income. The decisions must be made extremely quickly and should be close to optimal. The state of the art of online methods is presented in (Jaillet and Wagner, 2010) ; the solution time is the most important criterion. The problems studied can be modeled as allocation problems or vehicle routing problems, which are complex combinatorial problems. Online algorithms are designed to provide robust solutions regardless of the future events. Often the goal is to prove that it is impossible to obtain a solution worse than a certain bound.

There is an abundant literature on assignment problems, ranging from coupling problems (Karp et al., 1990) in a bipartite graph to more complex problems such as the Adwords problem (Mehta et al., 2007). Mehta et al. (2007) analyzed a greedy algorithm and more sophisticated methods for the Adwords problem. The primal-dual algorithm is applied in (Buchbinder, 2008) where fifteen different problems (including Adwords) are explored. One of the advantages of this algorithm is that it can update the selection parameters during the procedure to take past decisions into account. Recently, these algorithms have incorporated probabilistic information. To make better future decisions, Manshadi et al. (2012) use an offline strategy based on the available information before the beginning of the algorithm. Other methods using offline statistics are presented in (Karande et al., 2011; Jaillet and Lu, 2012). Several papers propose algorithms that use the available information when making decisions. Ciocan and Farias (2012) present a very general model that uses re-optimization techniques to incorporate this information. We use similar techniques to solve our problem of scheduling radiation therapy ; these techniques are outlined in (Legrain and Jaillet, 2013).

6.3 Methodology

The goal of the optimization is to determine the first day of treatment on a linac and then the slot. A linear program is used to schedule appointments on linacs. It must ensure the availability of sufficient resources to meet the patients' deadlines. Pretreatment (scanner, dosimetry, etc.) is performed before the radiotherapy treatment in a fixed amount of time depending on the priority of the patient.

Let $\mathcal{P}$ be the set of patients, $\mathcal{P}_c$ the set of curative patients, and $\mathcal{P}_p$ the set of palliative patients. We must schedule a set of treatments on linacs that have the following parameters for patient $j \in \mathcal{P}$:

— r_j : day when patient can start treatment (pretreatment is finished) ;

— d_j : deadline for first treatment ;

— p_j : treatment duration in days.

We first present an offline version of the problem for two reasons. The solution is optimal and therefore a lower bound for the online problem. Furthermore, the proposed model is used for the online procedure.

6.3.1 The Offline Model

We solve the problem for a known set of patients. The model is a set partitioning problem : we allocate each patient to a schedule while respecting the capacity of the linacs. A schedule consists of the treatment days for a patient and the cost of a schedule for a patient takes into account the number of waiting days and the number of days beyond its deadline. In our case, there are three deadlines : 3 days for palliative patients, 14 days for a specific type of curative patients, and 28 days for all other patients. Finally, the constraint on the ready date is a hard constraint.

More formally, let H be the index set of the working days given the planning horizon, $\mathcal{B}$ the index set of Mondays, and M the set of available linacs. Define S_j to be the index set of feasible schedules for patient j, a_{ijk}^m the plan $i \in S_j$ (= 1 if the patient is treated on machine

m on day k, and 0 otherwise), and c_{ij} the cost of the schedule $i \in S_j$. Let F_k^m be the number of available slots on linac m on day k, O_{day} the maximum number of overtime slots on linac m on day k, O_{week} the maximum number of overtime slots on linac m for a week, and c^o the cost of one overtime slot.
The variable x_{ij} defines the allocation of the plan $i \in S_j$ to patient j (=1 if allocated, and 0 otherwise). z_{mk} is the number of overtime slots on linac m on day k. This gives the following formulation for a known set of patients :

$$\min \sum_{j \in \mathcal{P}} \sum_{i \in S_j} c_{ij} x_{ij} + \sum_{k \in H} \sum_{m \in M} c^o z_{mk} \tag{6.1a}$$

subject to :

$$\sum_{i \in S_j} x_{ij} = 1, \qquad \forall j \in \mathcal{P} \tag{6.1b}$$

$$\sum_{j \in \mathcal{P}} \sum_{i \in S_j} a_{ijk}^m x_{ij} \leq F_k^m + z_{mk}, \qquad \forall m \in M, \forall k \in H \tag{6.1c}$$

$$\sum_{j \in \mathcal{P}_p} \sum_{i \in S_j} a_{ijk}^m x_{ij} \geq z_{mk}, \qquad \forall m \in M, \forall k \in H \tag{6.1d}$$

$$\sum_{k=b}^{b+4} z_{mk} \leq O_{week}, \qquad \forall m \in M, \forall b \in \mathcal{B} \tag{6.1e}$$

$$z_{mk} \in [0, O_{day}], \qquad \forall m \in M, \forall k \in H \tag{6.1f}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall j \in \mathcal{P}, \forall i \in S_j \tag{6.1g}$$

Constraint (6.1b) ensures that each patient j is scheduled. Constraint (6.1c) ensures that the capacity (including the overtime) of each linac is not exceeded. Constraint (6.1d) prevents curative patients from being scheduled in overtime slots. Constraints (6.1e) and (6.1f) bound respectively the weekly and daily overtime on each linac. Constraint (6.1g) is an integrality constraint. Finally, the objective (6.1a) is divided into two parts : the cost of a plan and the cost of overtime.

If all the plans are considered at the beginning, the computational time is too high. Therefore, the sets S_j of schedules are dynamically updated by column generation according to Algorithm 5.

The parameter δ is a constant and allows us to miss the deadline by a few days. This value ensures the feasibility of formulation (6.1) and should be small (e.g., less than a week). Algorithm 5 gives a nearly optimal solution : some schedules not generated by the procedure could improve the integer solutions. The plan i chosen for each patient j determines only

Algorithm 5 Offline Algorithm - Step 1

while there is a schedule with a negative reduced cost **do**
  solve linear relaxation of formulation (6.1)
  enumerate all feasible schedules for each patient j (start of treatment must be after day r_j and before day $d_j + \delta$)
  compute reduced costs of schedules from the dual variables
  insert schedules of patient j with a negative reduced cost into S_j, $\forall j \in \mathcal{P}$
end while
Solve formulation (6.1)

the first treatment day k and the linac m where the patient is treated. A slot must then be booked for each patient. The following theorem shows that a feasible solution always exists.

Theorem : For each feasible solution of formulation (6.1), there exists at least one feasible solution of the original problem. This solution can be built by Algorithm 6.

Algorithm 6 Offline Algorithm - Step 2

Sort all curative patients ($\mathcal{P}_c$) in ascending order of first day of treatment : let $\mathcal{R}$ be the resulting set ;
while $\mathcal{R}$ is not empty **do**
  consider the first patient $j \in \mathcal{R}$
  choose the first free slot p on the first treatment day k on the chosen linac m
  choose slot p on linac m for all treatment days
  remove patient j from $\mathcal{R}$
end while
Book palliative patients in the remaining free slots

This theorem also justifies the two-step approach : it shows that solving the overall problem is equivalent to solving the two problems sequentially.

Proof : First, we will prove by induction on $\mathcal{R}$ that the plan for the curative patients is feasible.

$\#\mathcal{R} = 0$ **:** There are no patients to schedule : the plan is feasible.

$\#\mathcal{R} = n+1$ **:** Suppose that the plan built with the constructive procedure is feasible for the first n patients. Let j be the last patient of $\mathcal{R}$. There exists a free slot p on the first treatment day k on linac m for patient j, because $\sum_{l \in \mathcal{P}_c} \sum_{i \in S_l} a^m_{ilk} x_{il} \leq F^m_k,\ \forall m \in M, \forall k \in H$. This inequality holds because $\sum_{l \in \mathcal{P}} \sum_{i \in S_l} a^m_{ilk} x_{il} \leq F^m_k + z_{mk}$ and $\sum_{l \in \mathcal{P}_p} \sum_{i \in S_l} a^m_{ilk} x_{il} \geq z_{mk}$. Slot p is free for all the subsequent days, because if a slot is not booked on day k, it will not be booked subsequently. A treatment occurs in the same slot on every day, and no patient currently booked starts treatment after day k. We therefore assign slot p on linac m to patient j. This plan is feasible.

By induction, the plan for all the curative patients is feasible.

Since there are enough free slots for all the palliative patients, the overall plan is also feasible. Consequently, if formulation (6.1) is solved to optimality, the overall solution will also be optimal.

6.3.2 The Online Procedure

Now we solve the model in an online fashion, i.e., the patients arrive sequentially and future patients are not known in advance. We present different algorithms for this problem.

Online algorithms

Two algorithms have often been used for online assignment problems : the greedy algorithm and the primal-dual algorithm. The first chooses the feasible plan with the lowest cost. We will use a slightly different algorithm : a minimum of b slots per linac per day are reserved for palliative patients. This number is set by the CICL. This policy ensures that sufficient slots are reserved for palliative patients while the remaining slots are allocated to curative patients. Indeed, at the CICL curative patients are scheduled at least one week before the beginning of their treatment while palliative patients are planned only a few days before. Algorithm 7 presents this modified greedy procedure. It is also the algorithm used by the CICL.

Algorithm 7 CICL Algorithm

```
for all patient arrivals j do
  if patient j is curative then
    choose first treament day k on linac m such that :
    1) the cost of the associated plan is minimal
    2) there are fewer than F_h^m − b slots booked on all treatment days
  else
    book palliative patient with the cheapest feasible plan
  end if
end for
```

Algorithm 7 finds the cheapest feasible plan in a greedy fashion, subject to a constraint on the minimal number of slots reserved for palliative patients. This procedure is also close to the

ASAP procedure. The longer the patient must wait, the more expensive the plan. Therefore, this algorithm will generally assign the first day of treatment to occur as soon as possible.

We use a primal-dual algorithm that draws inspiration from the examples of (Buchbinder, 2008). The algorithm makes an irrevocable decision at each patient arrival. The objective function is a simple maximize (or minimize) objective of the form $\sum_i (c_{ij} - r_i)x_{ij}$, where c_{ij} is a cost, r_i a constant dual variable, and x_{ij} the decision variable. Once the algorithm has made a choice, the dual variables are updated. Different primal-dual algorithms use different approaches to update the dual variables. In our algorithm, these variables are estimated at each patient arrival and are calculated using stochastic optimization tools.

Stochastic primal-dual algorithm

The stochastic optimization model is based on the offline formulation (6.1) and a sample set Ω_j of future scenarios. After the arrival of patient j, we consider the scenarios for future patients who arrive before day $d_j + p_j$. Let $\mathcal{P}^\omega$ be the set of these future patients for a scenario $\omega \in \Omega_j$. F_k^m is represents the remaining number of free slots (patients who arrived before j are already scheduled). O_{week} and O_{day} have been modified to take into account the palliative patients already booked in overtime. This gives the following formulation derived from formulation (6.1) to schedule patient j :

$$\min \sum_{i \in S_j} c_{ij} x_{ij} + \mathbb{E}_{\omega \in \Omega_j} [\sum_{l \in \mathcal{P}^\omega} \sum_{i \in S_l} c_{il} y_{il}^\omega + \sum_{k \in H} \sum_{m \in M} c^o z_{mk}^\omega]$$

subject to :

$$\begin{aligned}
&\sum_{i \in S_j} x_{ij} = 1 \\
&\sum_{i \in S_l} y_{il}^\omega = 1, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^\omega \\
&\sum_{i \in S_j} a_{ijk}^m x_{ij} + \sum_{l \in \mathcal{P}^\omega} \sum_{i \in S_l} a_{ilk}^m y_{il}^\omega \leq F_k^m + z_{mk}^\omega, \qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j \\
&\mathbb{1}_{\mathcal{P}_p}(j) \sum_{i \in S_j} a_{ijk}^m x_{ij} + \sum_{l \in \mathcal{P}_p^\omega} \sum_{i \in S_l} a_{ilk}^m y_{il}^\omega \geq z_{mk}^\omega, \qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j \\
&\sum_{k=b}^{b+4} z_{mk}^\omega \leq O_{week}, \qquad \forall m \in M, \forall b \in \mathcal{B}, \forall \omega \in \Omega_j \\
&z_{mk}^\omega \in [0, O_{day}], \qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j
\end{aligned} \tag{6.2}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in S_j$$
$$y_{il}^{\omega} \in \{0,1\}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in S_l, \forall \omega \in \Omega_j$$

For the details of the formula of the variables u_{ij}, see Appendix 6.5. To summarize, our current method can be stated as follows :

Algorithm 8 Stochastic Primal-Dual Algorithm

for all patient arrivals j **do**
  for all scenarios ω **do**
    solve dual slave problem (6.5) with value $x = 0$ for event ω
  end for
  compute the variables u_{ij} (as in Appendix 6.5)
  select a feasible schedule i that minimizes the cost $(c_{ij} + u_{ij})$
  update remaining resources F_k^m
end for

The reduced cost has two parts : c_{ij} is the current cost of the pattern, and u_{ij} is the expected cost of this pattern in the future. Algorithm 8 enumerates every feasible schedule to find the one that minimizes this reduced cost.

Online stochastic algorithm

Algorithm 8 chooses only the first day of treatment. We use another procedure to choose the slot, as in the offline case. When palliative patients arrive at the center, some curative patients have already been booked for the next seven days. Palliative patients simply fill the remaining slots of the plan and no optimization is necessary. Algorithm 8 is used only for curative patients and so $\mathbb{1}_{\mathcal{P}_p}(j) = 0$.
For a curative patient starting treatment on linac m on day k, we introduce the measure $G_{mk}(p)$ for a slot p. $G_{mk}(p)$ is equal to the number of consecutive free slots p just before day k on linac m. We choose the slot p with the minimum $G_{mk}(p)$. If a tie occurs, we choose the earliest slot in the day. We obtain Algorithm 9.

Algorithm 9 has three main benefits :

1. The remaining free slots for palliative patients are not constant but optimized according to the current and future workload ;
2. The ASAP procedure allows us to give palliative patients the earliest possible appointments ;
3. The procedure that chooses slot p tries to keep the largest slots for future patients.

Algorithm 9 Online Stochastic Algorithm

for all patient arrivals j **do**
  if patient j is curative **then**
    choose first treatment day k on linac m using Algorithm 8
    choose treatment slot p such that :
    $G_{mk}(p)$ is minimal and (if a tie occurs) p is minimal
  else
    book patient ASAP
  end if
end for

6.4 Results

We will first present the context of the results. Then, we will study the algorithms in a theoretical context to set some parameters. Finally, we will solve two real instances from the CICL.

6.4.1 Context of the results

We consider three types (t) of patients, each defined by its urgency, i.e., the deadline d_j for the start of treatment. Most urgent patients are palliative and must be treated within 3 days ; the lower-priority patients are curative and must be treated within 14 or 28 days. The center's main objective is to ensure that the patients meet their treatment deadlines. Our objective function has a penalty if a deadline is missed. Each penalty is modeled by a piecewise function : this is the most important part of the objective. Furthermore, the patient waiting time is slightly penalized to break the symmetry of the solution. Finally, the overtime is penalized to avoid unnecessary overtime. This penalty is difficult to set, because it depends on CICL policy, which tries to balance overtime and delayed treatments. We have set this penalty in such a way that overtime slots are booked only if a palliative patient misses his deadline by more than one day.

We compare three decision rules for scheduling patients : the CICL method (Algorithm 7), our online stochastic strategy (Algorithm 9) and the offline strategy (Algorithms 5 and 6). The CICL method reserves two free slots on each linac to accommodate possible future palliative patients (three-day deadline). The online stochastic algorithm uses scenarios that take into account all the unforeseen events that we want to model. The uncertainties that we consider are the number of patients, their arrival dates, their priorities, and their treatment durations. The offline strategy is almost optimal and represents the situation where all information is known in advance. Indeed, if every pattern is generated at the beginning, the optimality gaps are around 0.1%.

6.4.2 Sensitivity analysis

The scenarios are generated from a modified Poisson distribution. The number of each patient type arriving in a day follows a Poisson distribution truncated at a ceiling of twice the mean. For each patient type, the cancer type follows a Bernoulli distribution. The Poisson distributions are truncated to reduce the variability in the total number of patients. The parameters used were inferred from CICL historical data. The Poisson distribution is described by a total rate (λ) and the proportion (ρ_t) given in Table 6.1. Thus, for each patient type (t), the mean of the Poisson distribution is equal to $\rho_t\lambda$. The parameter ρ_t was inferred from CICL historical data. However, the arrival rate was not inferred, because it depends of the number of linacs. As the CICL is a new cancer treatment center, its number of linacs increased slowly from 2 to 6. To obtain reasonable results it is important to use an arrival rate λ that is representative of the number of linacs.

Table 6.1 – Proportions of patient types inferred from CICL data

Patient type t	palliative	curative 1 (14-day deadline)	curative 2 (28-day deadline)
Proportion ρ_t (%)	31	19	50

We allow 29 slots per linac per day and 3 overtime slots per linac per day. We allow a maximum of 5 overtime slots per linac per week. The online stochastic algorithm has been tested in two situations. The first uses the probability distribution : we refer to this as the online stochastic algorithm. The second uses both the probability distribution and all available information on low-priority patients, i.e., the patients with a 28-day deadline are known in advance : we refer to this as the online clairvoyant algorithm. We have assumed that all low-priority patients are known although, in reality, only 80% of them are registered at the CICL in advance.

Number of scenarios

The set Ω_j is normally very large, so we represent it by a few generated scenarios. We have to find a trade-off between the number of scenarios, the solution quality, and the computational time. We have tested different numbers of scenarios ($|\Omega_j|$) to study the behavior of the algorithms. We compare the value of the offline objective, the competitive ratio of the CICL algorithm, the competitive ratio of the online stochastic algorithm, and the ratio of the objectives of the CICL algorithm and the online stochastic algorithm. These are respectively : $Z^* = Obj_{offline}$, $c_{cicl} = \dfrac{Obj_{cicl}}{Obj_{offline}}$, $c_{online} = \dfrac{Obj_{online\ stochastic}}{Obj_{offline}}$, $c_{relative} = \dfrac{Obj_{cicl}}{Obj_{online\ stochastic}}$. The same ratios are defined for the online clairvoyant algorithm. We ran the tests 200 times and found the averages of these different measures. The planning horizon is 100 days, and the average number of patients is 168.5. We considered one linac with a moderate arrival rate λ of 1.7. We allowed a maximum of 5 s of solution time for each curative patient ; the CICL administrators consider this reasonable. All the tests were conducted with CPLEX 12.5 on an Intel(R) Xeon(R) X5675 (3.07 GHz) CPU.

Table 6.2 – Comparison of measures for different values of $|\Omega_j|$

$\|\Omega_j\| =$	1	2	3	5	10	15	20	25	30
Z^*	807776								
c_{cicl}	3.16								
Online Stochastic Algorithm									
Average c_{online}	3.17	2.92	2.80	2.79	2.70	2.73	2.72	2.78	2.71
Average $c_{relative}$	1.05	1.11	1.15	1.15	1.18	1.17	1.17	1.16	1.17
Average number of scenarios	1.0	2.0	3.0	5.0	10.0	14.8	19.1	22.8	25.9
Online Clairvoyant Algorithm									
Average c_{online}	2.32	2.24	2.17	2.16	2.13	2.12	2.10	2.10	2.10
Average $c_{relative}$	1.36	1.41	1.48	1.48	1.51	1.52	1.52	1.52	1.53
Average number of scenarios	1.0	2.0	3.0	5.0	10.0	15.0	19.8	24.4	28.8

The results presented in Table 6.2 are for the online stochastic algorithm and the online clairvoyant algorithm. For the former procedure, the use of available information improves the performance. As the number of scenarios increases, c_{online} decreases, stabilizing around 2.7. At the same time, $c_{relative}$ increases, stabilizing around 1.17. Note that these two ratios have apparently opposite behavior.

The online clairvoyant algorithm gives better results than the online stochastic procedure. The ratios follow the same trend as the number of scenarios increases : c_{online} decreases to 2.10 and $c_{relative}$ increases to 1.53. The use of the information greatly decreases the variability of the scenarios and has an immediate impact on the results.

Furthermore, the more scenarios the algorithms have, the better the ratios. However, more than 15 scenarios does not noticeably improve the results ; 10 scenarios seem to suffice.

Finally, when there are more than 20 scenarios, on average the online stochastic algorithm does not have enough time to solve all the scenarios. This is not the case for the clairvoyant algorithm. The problems of the online stochastic algorithm are harder to solve because the scenarios for a particular patient are more varied.

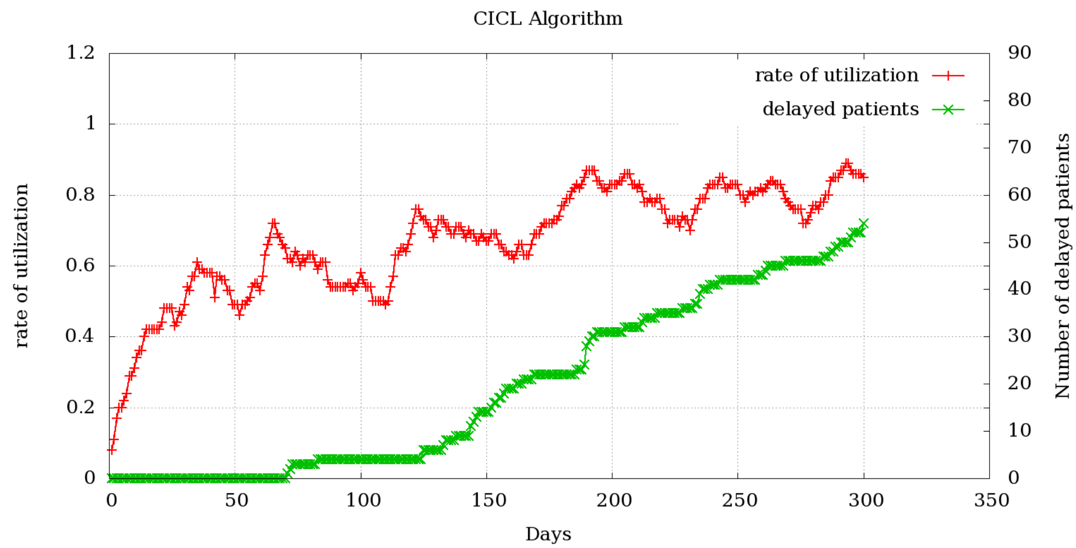
To summarize, the online stochastic and clairvoyant algorithms should generate 15 scenarios for the curative patients to achieve very good results. If we have good information about the future, the online clairvoyant algorithm can take advantage of it and perform much better than the CICL algorithm.
Since the CICL knows about 80% of the future low-priority curative patients, we will use the online clairvoyant algorithm in the following sections. Finally, if there is a tight constraint on the solution time, the algorithms will explore 15 scenarios ; otherwise they will use their 5 seconds for each curative patient.

Evolution of the algorithms

We study in this section the evolution of the CICL algorithm and the online clairvoyant procedure. We calculate the instantaneous rate of utilization by computing the average rate of utilization for a period of thirty days from the current day. This allows us to see the future workload. We want to measure the impact of a high or low rate on the behavior of the algorithms. We also study the evolution of the cumulative number of delayed patients per day. This cumulative function increases by one on a given day if a patient who has arrived on that day is scheduled after his deadline. The number of delayed patients increases quickly when the instantaneous rate is high. We also want to explore whether the online clairvoyant algorithm stabilizes the number of delayed patients and their waiting time compared to the CICL method. This could explain why the online clairvoyant algorithm outperforms the CICL procedure.
The planning horizon is 300 days, and there are 408 patients. The instance has been solved only once for one linac with an arrival rate λ of 1.5, and 15 scenarios have been used to find a schedule for each curative patient.

Until day 70 in Figure 6.1, there are no delayed patients because there are enough free slots on the linac to schedule all the patients before their deadlines. The instantaneous rate of utilization then starts increasing and goes up to 0.7. The number of delayed patients increases slightly for the CICL algorithm. The instantaneous rate decreases slightly and there are no delayed patients for either algorithm. The instantaneous rate then increases again. After this second rise, the number of delayed patients increases until the end of the time horizon : there are not enough slots to book new patients. The online clairvoyant algorithm increases in steps and not as quickly, and its rate of utilization has less variability. This algorithm obtains better results, because it stabilizes this rate and, consequently, it reserves a stable number of free slots for future patients.

There is a final difficulty for palliative patients. If a patient arrives on a Friday and is not



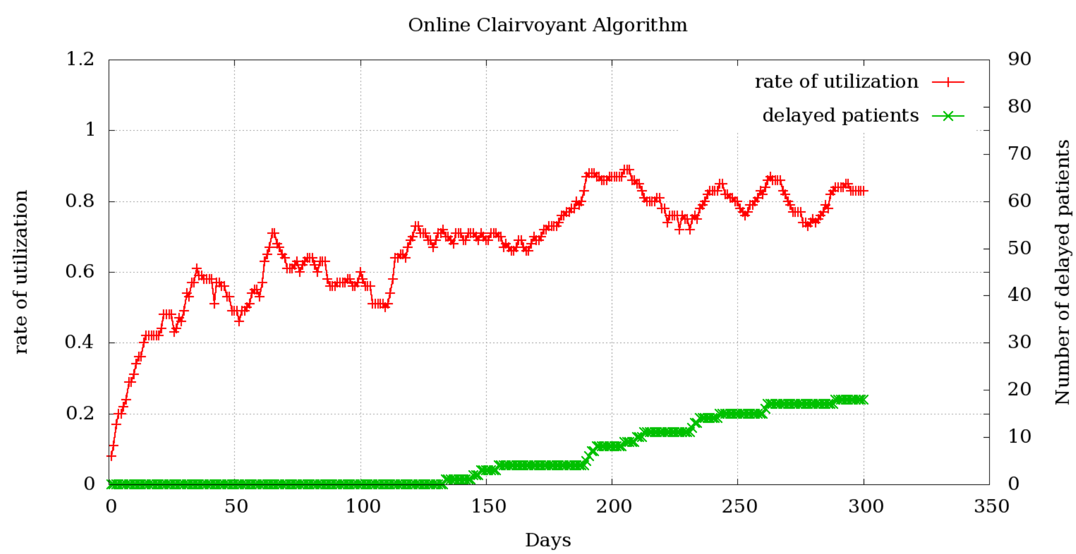


Figure 6.1 – Evolution of the number of delayed patients and the instantaneous rate of utilization

treated the same day, the first day of treatment is the subsequent Monday and the patient has already waited three days. Consequently, the patient has missed his deadline while waiting only one working day. Table 6.3 presents the distribution of delayed palliative patients by arrival day.

The CICL algorithm has scheduled more palliative patients after their deadlines. This is normal, because there are many delayed patients for this instance. However, these palliative patients are uniformly distributed across the week. For the clairvoyant algorithm, most of the delayed palliative patients arrive at the end of the week, and half of them arrive on Friday.

Table 6.3 – Distribution of delayed palliative patients by arrival day

	Arrival Day				
Algorithm	Monday	Tuesday	Wednesday	Thursday	Friday
CICL	4	6	8	6	8
Online Clairvoyant	0	1	2	2	5

If the weekend days were not counted toward the delay, this algorithm would have better results.

Objective function criteria

We present one instance with the value of the objective function as well as the number of delayed patients, the average delay, and the number of overtime slots. The aim is to check whether or not the objective function correctly measures the quality of the solution in a perfect situation, where the probability distribution is perfectly known. The planning horizon is 100 days and only one linac is used, as in Section 6.4.2.

Table 6.4 shows the results for the three strategies. The first set of columns gives for each strategy the number of patients who miss their deadlines, i.e., the waiting time is more than 3, 14, or 28 days. The next set of columns specifies the average waiting time for treatment, once again for each patient type. The final column shows the number of overtime slots. This instance has 140 patients, and the arrival rate λ is still 1.7. The online clairvoyant algorithm uses 15 scenarios for each curative patient. We obtain the following results : $Z^* = 494.5$, $c_{cicl} = 3.47$, $c_{online} = 1.33$, and $c_{relative} = 2.60$.

Table 6.4 – Results for a randomly generated instance

	Deadline			Average delay			Overtime slots
Algorithm	>3 days	>14 days	>28 days	3 days	14 days	28 days	
CICL	5	7	0	0.62	11.31	10.66	1
Online Clairvoyant	0	0	0	0.11	8.72	13.63	0
Offline	0	0	0	0.0	7.31	10.08	0

In terms of solution quality, the results presented in Table 6.4 are satisfactory. They show that the online clairvoyant procedure meets the needs of booking policies better than the strategy typically used in cancer centers : 5 palliative patients (3.6%) and 7 curative patients (5.0%) with a 14-day deadline are better served.

In terms of waiting times, patients with a 14-day deadline have an average delay of 8.72 days, compared to 11.31 days with the CICL strategy. However, those with a 28-day deadline have an average delay of 13.63 days. The distribution of these delays is better aligned with the original deadlines, i.e., those with a later deadline tend to wait more. The average deadlines obtained by the online clairvoyant and the offline algorithms compare very well : the two solutions have roughly the same average waiting time. It is slightly higher for the online clairvoyant procedure ; it is because this algorithm tries to build in some flexibility for future patients whereas the offline algorithm knows the patients in advance.

To summarize, the offline solution is almost optimal and all the deadlines are met in this example. The online clairvoyant approach compares very well to the offline solution : no patients miss their deadlines, and the average waiting time is comparable to that of the offline solution. The CICL approach is clearly the simplest since it involves no computation, but 12 patients miss their deadlines, and the average waiting time is no better than that of the online clairvoyant algorithm.

All the measures also indicate that the offline algorithm provides the best solution : the ratios c_{cicl} and c_{online} are greater than 1. This solution is optimal and can only be achieved in a perfect world where the future patients are completely known. Furthermore, the ratio c_{online} is close to 1, indicating that the online solution is close to the offline solution, despite its imperfect knowledge of the future. Finally, we note that the ratios are better than their averages (refer to Table 6.2), but the differences between the CICL method and the online clairvoyant algorithm are large. To conclude, the ratios c_{cicl} and $c_{relative}$ confirm that the CICL solution is the worst in this case.

6.4.3 Real instances

This section presents two real CICL instances with 74 and 77 working days. We ran the tests with 2 linacs, 23 slots per day for the first instance and 29 for the second, and 3 overtime slots per linac per day for both. We allowed a maximum of 5 overtime slots per linac per week. The first instance books 159 patients and the second 181. We generated future patients with an arrival rate λ of 2.75 for the first instance and 3.5 for the second. The tests were run 200 times for each instance with 15 scenarios and a maximum of 5 s of solution time, but only one run is presented in Tables 6.5 and 6.6.

Table 6.5 – 159 patients in 74 working days

	Deadline			Average delay			Overtime slots
Algorithm	>3 days	>14 days	>28 days	3 days	14 days	28 days	
CICL	6	16 (11)	0	1.11	16.32	14.23	1
Online Clairvoyant	4	6	0	0.57	10.24	16.03	2
Offline	0	0	0	0.0	7.12	14.42	0

The solution presented in Table 6.5 has the following values : $c_{online} = 3.07$ (for a mean of 2.82 and a standard deviation of 0.17), $c_{relative} = 11.91$ (for a mean of 13.33 and a standard deviation of 0.98), and 14.3 scenarios on average. This performance is slightly worse than the average. However, the solution obtained for the online clairvoyant algorithm outperforms the CICL solution. In particular, the CICL solution delays 16 curative patients including 11 high-priority curative patients who have waited more than 21 days. However, the low-priority curative patients wait in average two more days for the online clairvoyant solution. The improvement of the global solution penalizes these patients, but they are still treated before their deadline. Finally, the offline solution delays no patients and the high-priority palliative and curative patients are nearly served at their ready date.

Table 6.6 – 181 patients in 77 working days

	Deadline			Average delay			Overtime slots
Algorithm	>3 days	>14 days	>28 days	3 days	14 days	28 days	
CICL	15	4	0	1.78	9.19	8.65	13
Online Clairvoyant	5	0	0	0.78	8.54	9.98	0
Offline	0	0	0	0.0	7.13	10.14	0

The solution presented in Table 6.6 has the following values : $c_{online} = 2.42$ (for a mean of 1.74 and a standard deviation of 0.13), $c_{relative} = 4.08$ (for a mean of 5.76 and a standard deviation of 0.95), and 55.4 scenarios on average. These ratios are still worse than the average, and the online clairvoyant algorithm continues to perform better than the CICL procedure. The online clairvoyant algorithm keeps delaying low-priority curative patients while high-priority patients have a better access to the center. Finally, the online clairvoyant and offline results are similar for this instance.

Conclusion

We have presented the problem of scheduling patient appointments in an online fashion. The goal is to offer patients a reasonable waiting time for their first treatment, while maximizing the resource utilization, and the ultimate goal is to provide better access to treatment. Curative patients are offered regular appointments, i.e., at the same time every treatment day, to accommodate their other commitments. Palliative patients are offered varying appointments. Booking curative patients has a clear impact on the utilization of the machines since they might not be available for future palliative patients. An offline two-step algorithm provides an optimal solution when all patients are known.

To reflect the uncertainty related to the arrival of patients and the treatment duration, we developed an approach combining stochastic optimization and online optimization. This approach is innovative and our contributions are in both the methodology and the application. The results show that this approach works well on real instances and outperforms the current strategy.

In future work we will schedule the upstream flow. Our current formulation assumes that the pretreatment takes a constant numbers of days whatever the workload. We want to estimate this number of days more accurately to avoid delays. If the pretreatment is not completed on time, the patient cannot start radiotherapy and the corresponding slots should be canceled. Instead of enumerating all feasible plans, we will solve a more complex problem that takes into account the pretreatment and continues to minimize the reduced costs. This research will lead to a way to solve more complex stochastic column generation problems.
We are currently collaborating with the CICL and the software provider to pilot the project and integrate the algorithm to the software.

6.5 The online stochastic algorithm

We present here the online stochastic algorithm introduced by Legrain and Jaillet (2013). We rewrite this algorithm for our application. Recall the stochastic optimization formulation (6.2) :

$$\min \sum_{i \in S_j} c_{ij} x_{ij} + \mathbb{E}_{\omega \in \Omega_j} [\sum_{l \in \mathcal{P}^\omega} \sum_{i \in S_l} c_{il} y^\omega_{il} + \sum_{k \in H} \sum_{m \in M} c^o z^\omega_{mk}]$$

subject to :

$$\begin{aligned}
&\sum_{i \in S_j} x_{ij} = 1 \\
&\sum_{i \in S_l} y^\omega_{il} = 1, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^\omega \\
&\sum_{i \in S_j} a^m_{ijk} x_{ij} + \sum_{l \in \mathcal{P}^\omega} \sum_{i \in S_l} a^m_{ilk} y^\omega_{il} \leq F^m_k + z^\omega_{mk}, \\
&\qquad\qquad\qquad\qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j \\
&\mathbb{1}_{\mathcal{P}_p}(j) \sum_{i \in S_j} a^m_{ijk} x_{ij} + \sum_{l \in \mathcal{P}^\omega_p} \sum_{i \in S_l} a^m_{ilk} y^\omega_{il} \geq z^\omega_{mk}, \\
&\qquad\qquad\qquad\qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j \\
&\sum_{k=b}^{b+4} z^\omega_{mk} \leq O_{week} \qquad \forall m \in M, \forall b \in \mathcal{B}, \forall \omega \in \Omega_j
\end{aligned}$$

$$
\begin{aligned}
& z^{\omega}_{mk} \in [0, O_{day}], \qquad \forall m \in M, \forall k \in H, \forall \omega \in \Omega_j \\
& x_{ij} \in \{0,1\}, \qquad \forall i \in S_j \\
& y^{\omega}_{il} \in \{0,1\}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in S_l, \forall \omega \in \Omega_j
\end{aligned}
$$

6.5.1 L-shaped method

The L-Shaped method (Birge and Louveaux, 2011) enables us to solve formulation (6.2). This technique is based on Benders decomposition (Benders, 1962). Formulation (6.2) is first decomposed into a master problem (6.3) and slave problems (6.4).

$$
\text{(Master problem)} \ \min \sum_{i \in S_j} c_{ij} x_{ij} + \mathbb{E}_{\omega \in \Omega_j}[Q(x,\omega)]
$$

subject to :

$$
\begin{aligned}
& \sum_{i \in S_j} x_{ij} = 1 \\
& x_{ij} \in \{0,1\}, \qquad \forall i \in S_j
\end{aligned} \tag{6.3}
$$

The function $Q(x, \omega)$ is called the recourse ; it deals with the stochastic part of the objective. The goal of the slave problems (6.4) is to compute the value of this function for each x and each ω. The variables inside the parentheses (.) are the dual variables associated with the constraint.

$$
\text{(Slave problems)} \ Q(x,\omega) \ = \ \min \sum_{l \in \mathcal{P}^{\omega}} \sum_{i \in S_l} c_{il} y^{\omega}_{il} + \sum_{k \in H} \sum_{m \in M} c^{o} z^{\omega}_{mk}
$$

subject to :

$$
\begin{aligned}
& \sum_{i \in S_l} y^{\omega}_{il} = 1, \qquad \forall l \in \mathcal{P}^{\omega} \ (\alpha^{\omega}_{l}) \\
& \sum_{l \in \mathcal{P}^{\omega}} \sum_{i \in S_l} a^{m}_{ilk} y^{\omega}_{il} \leq F^{m}_{k} + z^{\omega}_{mk} - \sum_{i \in S_j} a^{m}_{ijk} x_{ij}, \qquad \forall m \in M, \forall k \in H \ (\beta^{\omega}_{mk}) \\
& \sum_{l \in \mathcal{P}^{\omega}_{p}} \sum_{i \in S_l} a^{m}_{ilk} y^{\omega}_{il} \geq z^{\omega}_{mk} - \mathbb{1}_{\mathcal{P}_p}(j) \sum_{i \in S_j} a^{m}_{ijk} x_{ij}, \qquad \forall m \in M, \forall k \in H \ (\gamma^{\omega}_{mk})
\end{aligned} \tag{6.4}
$$

$$\sum_{k=b}^{b+4} z^{\omega}_{mk} \leq O_{week}, \qquad \forall m \in M, \forall b \in \mathcal{B}\ (\pi^{1\omega}_{mb})$$
$$z^{\omega}_{mk} \in [0, O_{day}], \qquad \forall m \in M, \forall k \in H\ (\pi^{2\omega}_{mk})$$
$$y^{\omega}_{il} \in \{0,1\}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in S_l\ (\pi^{3\omega}_{mk})$$

Since the recourse function Q has to be computed for each value of its variables, an approximation of Q is built. First, we relax the integrality constraint on the variables y^{ω}_{il} and then we use the duals (6.5) of the slave problems.

$$\text{(Dual slave problems)}\ \max \sum_{l\in\mathcal{P}^{\omega}} \alpha^{\omega}_l - \sum_{m\in M}\{\sum_{k\in H}[(F^m_k - \sum_{i\in S_j} a^m_{ijk}x_{ij})\beta^{\omega}_{mk} + \\ \mathbb{1}_{\mathcal{P}_p}(j)(\sum_{i\in S_j} a^m_{ijk}x_{ij})\gamma^{\omega}_{mk} + O_{day}\pi^{2\omega}_{mk} + \pi^{3\omega}_{mk}] + \sum_{b\in\mathcal{B}} O_{week}\pi^{1\omega}_{mb}\}$$

subject to :

$$\begin{aligned}
&\alpha^{\omega}_l + \sum_{m\in M}\sum_{k\in H} a^m_{ilk}(\gamma^{\omega}_{mk} - \beta^{\omega}_{mk}) - \pi^{3\omega}_{mk} \geq c_{il}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in S_l\\
&\beta^{\omega}_{mk} - \gamma^{\omega}_{mk} - \pi^{1\omega}_{mb(k)} - \pi^{2\omega}_{mk} \geq c^o, \qquad \forall m \in M, \forall k \in H\\
&\beta^{\omega}_{mk},\ \gamma^{\omega}_{mk},\ \pi^{2\omega}_{mk},\ \pi^{3\omega}_{mk} \geq 0, \qquad \forall m \in M, \forall k \in H\\
&\pi^{1\omega}_{mb} \geq 0, \qquad \forall m \in M, \forall b \in \mathcal{B}
\end{aligned} \tag{6.5}$$

where $b(k)$ is the index of the Monday of the same week as the working day indexed by k. These problems are always feasible ; furthermore they are bounded when there are enough free slots on the linacs for the chosen pattern i ($x_{ij} = 1$). There always exists such a pattern, because the first day of treatment can be sufficiently far from the current day to ensure that there are enough free slots.
Thus, the weak duality theorem gives an approximation (a cut) of the recourse function Q :

$$\forall i \in S_j, \forall\omega \in \Omega_j,\ Q(x,\omega) \geq \sum_{l\in\mathcal{P}^{\omega}} \alpha^{\omega}_l - \sum_{m\in M}\{\sum_{k\in H}[F^m_k\beta^{\omega}_{mk} + \sum_{i\in S_j} a^m_{ijk}x_{ij}(-\beta^{\omega}_{mk} + \mathbb{1}_{\mathcal{P}_p}(j)\gamma^{\omega}_{mk}) + \\ O_{day}\pi^{2\omega}_{mk} + \pi^{3\omega}_{mk}] + \sum_{b\in\mathcal{B}} O_{week}\pi^{1\omega}_{mb}\} \tag{6.6}$$

For the qth cut for scenario ω, approximation (6.6) can be simplified. All the constants of this cut can be gathered into one constant C_q^ω :

$$C_q^\omega = \sum_{l\in\mathcal{P}^\omega} \alpha_l^\omega - \sum_{m\in M}\{\sum_{k\in H}[F_k^m\beta_{mk}^\omega + O_{day}\pi_{mk}^{2\omega} + \pi_{mk}^{3\omega}] + \sum_{b\in\mathcal{B}} O_{week}\pi_{mb}^{1\omega}\}$$

Each time the slave problems are solved, we obtain a cut for the master problem (6.3). The master problem is then transformed as follows :

$$\text{(Master problem)} \qquad \min \sum_{i\in S_j} c_{ij}x_{ij} + \mathbb{E}_{\omega\in\Omega_j}[\theta^\omega] \tag{6.7a}$$

subject to :

$$\sum_{i\in S_j} x_{ij} = 1$$

$$\theta^\omega \geq C_q^\omega + \sum_{i\in S_j}(\sum_{m\in M}\sum_{k\in H} a_{ijk}^m(\beta_{mk}^\omega - \mathbb{1}_{\mathcal{P}_p}(j)\gamma_{mk}^\omega))x_{ij}, \qquad \forall\omega\in\Omega_j, \forall q \tag{6.7b}$$

$$x_{ij}\in\{0,1\}, \qquad \forall i\in S_j$$

Benders Decomposition is also known as row generation. The duals (6.5) of the slave problems generate cuts that are inserted into the master problem (6.7). The L-Shaped procedure (Algorithm 10) uses this decomposition.

Algorithm 10 L-Shaped procedure

$q = 0,\ x^q = 0, x = 1$
while $x^q \neq x$ **do**
  $x = x^q$
  $q{+}{+}$
  for all $\omega \in \Omega_j$ **do**
    solve the dual slave problem with the value x for the event ω
    add the qth cut to the master problem (6.7)
  end for
  solve the master problem (6.7) and save the solution in x^q
end while

When the L-Shaped procedure stops, the optimum is reached. Indeed, when the dual slave problems are solved for a value x, the recourse function Q is equal to the cut in this point thanks to the strong duality theorem. Then, if the master problem (6.7) finds the same solution x, it has to be the optimum, because there is no approximation at this point and all the other points are underestimated.

6.5.2 Stochastic primal-dual algorithm

Legrain and Jaillet (2013) make a simplification : the master problem is solved only once. Then, for each event ω, there is only one cut inserted into the master problem (6.7). Thus, the constraints (6.7b) become equalities due to the minimization. The variables θ are therefore replaced in the objective (6.7a) by their expressions. The objective (6.7a) becomes : $\min \sum_{i \in S_j} c_{ij} x_{ij} + \mathbb{E}_{\omega \in \Omega_j}[C_q^\omega + \sum_{i \in S_j} (\sum_{m \in M} \sum_{k \in H} a_{ijk}^m (\beta_{mk}^\omega - \mathbb{1}_{\mathcal{P}_p}(j) \gamma_{mk}^\omega)) x_{ij}]$.
We can also remove the constants C_0^ω from the objective to obtain the final master problem (6.8) :

$$\text{(Master problem)} \ \min \sum_{i \in S_j} (c_{ij} + \sum_{m \in M} \sum_{k \in H} \mathbb{E}_{\omega \in \Omega_j}[\beta_{mk}^\omega - \mathbb{1}_{\mathcal{P}_p}(j) \gamma_{mk}^\omega] a_{ijk}^m) x_{ij}$$

subject to :

$$\begin{aligned} & \sum_{i \in S_j} x_{ij} = 1 \\ & x_{ij} \in \{0, 1\}, \qquad \forall i \in S_j \end{aligned} \tag{6.8}$$

Finally, the master problem (6.8) just chooses the pattern with the minimum objective. We can link our procedure with the primal-dual algorithm. The two algorithms are similar ; the only difference is the computation of the variables r_i. If we define the variables $u_{ij} = \sum_{m \in M} \sum_{k \in H} \mathbb{E}_{\omega \in \Omega_j}[\beta_{mk}^\omega - \mathbb{1}_{\mathcal{P}_p}(j) \gamma_{mk}^\omega] a_{ijk}^m$, we obtain the stochastic primal-dual algorithm (8).

CHAPITRE 7 ARTICLE 4 : COMBINING BENDERS AND DANTZIG-WOLFE DECOMPOSITIONS FOR ONLINE STOCHASTIC COMBINATORIAL OPTIMIZATION

A. Legrain, N. Lahrichi, LM. Rousseau et M. Widmer ont écrit cet article et l'ont soumis en 2015 à *Operations Research*.

7.1 Introduction

In resource allocation problems, the operator decides how to allocate requests to resources in order to maximize the profit or improve the service quality of an organization. These problems are challenging because the operator must quickly and continuously make irrevocable allocations without full knowledge of future requests. However, with the spread of information systems, there is now much historical data available to support online decisions. Many operators would benefit from decision support systems that use forecasts and optimization to guide allocation decisions.

Online resource allocation problems have been widely studied in the past 15 years and arise in domains such as :

- **Search-engine advertisements.** A search-engine operator allocates in an online fashion an advertisement to each new keyword search ; it is displayed in the users' web navigator window. The goal is to maximize across all the keywords the expected advertisement revenue without exceeding the limited budget of each advertiser.
- **Revenue management.** Companies such as airlines that sell a limited quantity of goods match each new purchase to a selling price. They seek to maximize their expected revenue by choosing their selling prices dynamically without full knowledge of the demand.
- **Appointment booking.** Healthcare organizations such as clinics set up daily appointments with medical personnel and/or resources. The operator must find a compromise between efficiency and waiting-time targets that are based on patient priority.
- **Vehicle routing.** In a given time horizon, a fleet of vehicles must serve customers that are either known in advance or revealed dynamically. New customers must be added to the scheduled routes in real time.
- **Task assignment.** New tasks are generally assigned to workers according to their priority. The quality of the schedule must be continously reoptimized over a given horizon to ensure efficiency.

In an ideal (deterministic) world, the operator seeks a sequence of decisions leading to an optimal solution. However, in a realistic (stochastic) environment, not even feasibility can be ensured, because the operator makes irrevocable decisions without full knowledge of the future requests. The operator instead tries to minimize the expected deviation from the optimal solution based on the forecasts, which may be dynamically updated during the planning horizon.

The state-of-the-art methods for online resource allocation problems fail to evaluate the expected impact of allocation decisions. If the probability distribution of the requests is known at the beginning of the process and remains unchanged, approximate dynamic programming (Powell, 2007) can be used to compute an offline policy. Online optimization (Buchbinder, 2008) provides efficient algorithms with a proof of competitiveness, but it does not take advantage of available forecasts. Finally, online stochastic (OS) optimization (Van Hentenryck and Bent, 2009) is a broad and practical paradigm, but it does not provide general mathematical tools to approximate the impact of allocation decisions.

In this paper, we extend the general framework of OS algorithms to provide such tools. We use Benders (Benders, 1962) and Dantzig–Wolfe (Dantzig and Wolfe, 1960) decompositions to solve OS resource allocation problems. The advantages of this new framework are threefold.

1. Benders subproblems are used to estimate the feasibility and optimality of the final solution. These estimations are handled through a given primal-ratio in the spirit of chance constraints, thus accommodating the operator's level of risk aversion.
2. Benders subproblems also allow us to infer the future load on the resources. A dual variable associated with each resource gives an approximation of the future cost of one resource unit.
3. Large-scale problems can be solved, because the combinatorial explosion and the complex constraints are managed through Dantzig–Wolfe decomposition.

We illustrate the use of this general framework on two specific applications that are representative of the complexity of online resource allocation problems. The first is an appointment booking and scheduling problem in a cancer treatment facility. Patients with varying priorities must be allocated consecutive treatment sessions on linear accelerators (linacs). The algorithm minimizes the patients' waiting time and schedules the necessary treatment-preparation steps. The second application concerns task assignment and routing decisions in a warehouse, where there is a queue of prioritized tasks that arrive continuously. When a worker finishes the current route, the operator must assign a new one. The algorithm must ensure the feasibility and efficiency of the assigned route while taking into account the priority ordering of the tasks.

The rest of the paper is organized as follows. Section 7.2 reviews existing techniques for online resource allocation problems. Section 7.3 formally defines the problem, and Section 7.4 details the algorithm. Section 7.5 discusses the appointment booking and scheduling application, while Section 7.6 presents the task assignment and routing application. Sections 7.5 and 7.6 both provide computational results for simulated and real data. Section 7.7 provides concluding remarks.

7.2 Literature review

Three main strategies have been proposed for online resource allocation problems : computing an offline policy, following a simple online policy, or reoptimizing the system for each request.

Markov decision processes (Puterman, 2014) can be used to compute an offline policy. The problem is decomposed into two different sets (states and actions) and two functions (transition and reward). A state describes the value of the resources, and an action represents an available decision. The transition function indicates the probability of reaching one state from another with an action, and the reward function gives the reward for applying an action from a given state. An offline policy is then computed for each state. Approximate dynamic programming (Powell, 2007) proposes ways to deal with the curse of dimensionality created by the exponential growth of the size of the state space. This technique has been successfully applied to financial optimization (Bäuerle and Rieder, 2011), booking (Patrick et al., 2008a), and routing (Novoa and Storer, 2009) problems. However, the main advantage of this technique is also its main drawback : it concentrates all the computation at the beginning of the process, and then the operator has to follow the policy. This approach works only with a distribution known a priori ; otherwise the transition function must be updated at each stage, thus negating the effort invested in the initial computation.

Online algorithms aim to solve dynamic problems rapidly without any knowledge of future requests. They ensure the quality of the final solution via a competitive ratio, which measures the gap between the optimal and worst-case solutions. Karp et al. (1990) solve the online matching problem. Mehta et al. (2007) introduce the Adwords problem and propose a $(1 - \frac{1}{e})$-competitive algorithm. Buchbinder (2008) proposes a primal-dual algorithm for a wide range of problems such as set covering, routing, and resource allocation problems. Feldman et al. (2009); Karande et al. (2011); Manshadi et al. (2012), and Jaillet and Lu (2014) introduce probabilistic knowledge. They compute offline strategies based on a prior distribution to help the online algorithm. Feldman et al. (2010) and Jaillet and Lu (2012) go further by reoptimizing their policy with the information available after one stage. However, these authors all consider the matching problem, which is a special polynomial case of the resource allocation problem. Legrain and Jaillet (2013) present a reoptimized primal-dual algorithm for the search-engine advertisement problem (i.e., the Adwords problem). Ciocan and Farias (2012) propose an OS algorithm for bipartite resource allocation problems. They compute an offline policy based on a prior distribution and reoptimize it at each time step. The policy estimates how to distribute each type of request among the different feasible allocations. There must be high volumes of each type of request in order for the algorithm to converge to the expected distribution. These authors do not provide a competitiveness proof for complex resource allocation problems, and they do not present generic techniques for using available forecasts of future requests.

OS algorithms reoptimize the problem for each new request using up-to-date forecasts. Classical techniques such as stochastic programming (Birge and Louveaux, 2011) are not well suited to online optimization because they are too time-consuming ; see Powell and Roy (2004). Van Hentenryck and Bent (2009) provide a general framework for OS problems and give three algorithms : expectation, consensus, and regret. These algorithms are all based on procedures that solve the offline version of the problem. The regret algorithm is the most advanced. In this algorithm, there are three steps to perform for each new request. First, build sample scenarios of future requests, then solve each scenario once with an offline procedure, and finally make a heuristic decision based on the solutions found. Our OS algorithm is based on this procedure. We propose a mathematical programming scheme for resource allocation problems to analyze the solution of each scenario and make the best decision for each request.

7.3 Problem formulation

We now give a formal description and formulation of the online resource allocation problem. A total of T requests arrive one at a time during a time horizon H. The operator must allocate the jth request so as to minimize the objective function. There is a set $\mathcal{R}$ of consumable resources, and resource r has b_r units available.

7.3.1 Offline formulation

We describe each allocation via a resource consumption pattern as in a Dantzig–Wolfe decomposition : each allocation pattern $i \in \mathcal{S}_j$ with cost c_{ij} for the jth request consumes an amount A_{ijr} of resource r. When the set $\mathcal{S}_j$ is large, the allocation patterns are generated during the solution process via column generation (Desaulniers et al., 2005). Complex and operational constraints (e.g., time constraints for the scheduling) are hidden in the patterns, giving a simple offline formulation for the resource allocation problem. Note that these constraints do not induce an integrality gap, thus strengthening the linear relaxation.

$$\min \sum_{j=1}^{T} \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} \tag{7.1}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1, \qquad \forall j = 1 \ldots T \tag{7.2}$$

$$\sum_{j=1}^{T} \sum_{i \in \mathcal{S}_j} A_{ijr} x_{ij} \leq b_r \qquad \forall r \in \mathcal{R} \tag{7.3}$$

$$x_{ij} \in \{0, 1\} \qquad \forall j = 1 \ldots T, \forall i \in \mathcal{S}_j \tag{7.4}$$

The variable x_{ij} is 1 if the jth request is matched to allocation pattern i, and 0 otherwise. The objective (7.1) minimizes the total cost of the allocations. Constraints (7.2) ensure that each request is matched to one allocation pattern. Constraints (7.3) manage the resource consumption. We now extend the offline formulation to the OS problem.

7.3.2 Online stochastic formulation

In a dynamic environment, the resource allocation problem becomes a multistage problem. One way to handle the allocation decision for the jth request is via a two-stage program with fixed recourse. Classical stochastic tools use a scenario-based optimization to solve this program. The number of requests T and the nature of each request are not known in advance and must be determined for each scenario. If the horizon H is known and sufficient historical data is available to build a probability distribution, which can be empirical, scenarios of future requests can be sampled through this probability distribution.

Let the sample set Ω_j be the set of possible scenarios of future requests. Each scenario ω has a probability p^ω and a total number of requests T^ω. The variable y_{il}^ω with cost c_{il}^ω is 1 if the lth request of scenario ω is matched to allocation pattern i, and 0 otherwise. The following stochastic formulation chooses the allocation of the jth request.

$$\min \sum_{i \in \mathcal{S}_j} c_{ij} x_{ij} + \sum_{\omega \in \Omega_j} p^\omega \sum_{l=1}^{T^\omega} \sum_{i \in \mathcal{S}_l^\omega} c_{il}^\omega y_{il}^\omega \tag{7.5}$$

subject to

$$\sum_{i \in \mathcal{S}_j} x_{ij} = 1 \tag{7.6}$$

$$\sum_{i \in \mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall \omega \in \Omega_j, \forall l = 1 \ldots T^{\omega} \tag{7.7}$$

$$\sum_{i \in \mathcal{S}_j} A_{ijr} x_{ij} + \sum_{l=1}^{T^{\omega}} \sum_{i \in \mathcal{S}_l} A_{ilr} y_{il}^{\omega} \leq b_r, \qquad \forall \omega \in \Omega_j, \forall r \in \mathcal{R} \tag{7.8}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in \mathcal{S}_j \tag{7.9}$$

$$y_{il}^{\omega} \in \{0, 1\}, \qquad \forall \omega \in \Omega_j, \forall l = 1 \ldots T^{\omega}, \forall i \in \mathcal{S}_l \tag{7.10}$$

The objective (7.5) minimizes the allocation cost of the jth request plus the expected cost of future allocations. Constraints (7.6) and (7.7) ensure that each (current or future) request is matched to one allocation pattern. Constraints (7.8) continue to manage the global resource consumption for each scenario. Constraints (7.9) and (7.10) define x_{ij} and y_{il}^{ω} as binary variables. This formulation leads to a huge model, which is difficult to solve in an online time-limited environment.

7.4 Methodology

We propose a general, fast, and efficient L-shaped-based algorithm to rapidly solve the allocation problem for the jth request. Our procedure minimizes the total expected cost of future allocations in order to make the best decision for the current allocation. It first computes an expected descent direction using the value of the dual variables associated with the resource constraints (7.8). When this descent direction is imprecise for some resource constraints, we restrict the search space by generating probabilistic cuts for these constraints.

We first introduce the classical L-shaped procedure (Slyke and Wets, 1969) for the online resource allocation problem. It applies Benders decomposition (Benders, 1962) to transfer all the stochastic components (parts of the objective (7.5) and constraints (7.7), (7.8), and (7.10)) into an integer subproblem for each scenario ω.

$$Q(\overline{x}_{ij}, \omega) = \min \sum_{l=1}^{T^{\omega}} \sum_{i \in \mathcal{S}_l^{\omega}} c_{il}^{\omega} y_{il}^{\omega} \tag{7.11}$$

subject to

$$\sum_{i\in\mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall l = 1\ldots T^{\omega} \qquad (\alpha_l^{\omega}) \tag{7.12}$$

$$\sum_{l=1}^{T^{\omega}}\sum_{i\in\mathcal{S}_l} A_{ilr} y_{il}^{\omega} \leq b_r - \sum_{i\in\mathcal{S}_j} A_{ijr}\overline{x}_{ij}, \qquad \forall r \in \mathcal{R} \qquad (\beta_r^{\omega}) \tag{7.13}$$

$$y_{il}^{\omega} \in \{0,1\}, \qquad \forall l = 1\ldots T^{\omega}, \forall i \in \mathcal{S}_l \tag{7.14}$$

The dual variables α_l^{ω} and β_r^{ω} (in parentheses) are associated with constraints (7.12) and (7.13). The integer subproblem calculates the recourse function $Q(\overline{x}_{ij}, \omega)$ for a solution $\overline{x}_{ij}$ and a scenario ω. We solve the linear relaxation of the subproblem because a two-stage recourse problem with integer subproblems is much more difficult. The solution of each relaxed subproblem gives the load on the resources : the dual variables β_r^{ω} give the expected future cost of one unit of resource r.

The relaxed subproblem feeds the following master problem for each scenario ω with an optimality cut (7.17), which approximates the recourse function $Q(x_{ij}, \omega)$:

$$\min \sum_{i\in\mathcal{S}_j} c_{ij} x_{ij} + \sum_{\omega\in\Omega_j} p^{\omega}\theta^{\omega} \tag{7.15}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1 \tag{7.16}$$

$$\theta^{\omega} \geq \sum_{r\in\mathcal{R}} \beta_r^{\omega}(b_r - \sum_{i\in\mathcal{S}_j} A_{ijr} x_{ij}) + \sum_{l=1}^{T^{\omega}} \alpha_l^{\omega}, \qquad \forall \omega \in \Omega_j \tag{7.17}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{7.18}$$

The L-shaped procedure iteratively solves the master problem and the relaxed subproblems, which add new cuts to the master. The procedure stops when the relaxed subproblems have already been solved for the current solution of the master problem (i.e., the relaxed subproblems will not generate new optimality cuts).

In a dynamic environment, the L-shaped procedure is too slow. We therefore propose a much faster one-iteration L-shaped procedure.

7.4.1 One-iteration L-shaped procedure

If just one iteration is performed, the optimality cuts (7.17) can be transferred without the constant parts in the objective (7.19), which is now decomposed into two parts : the real cost c_{ij} of allocation pattern i and the average negative cost $\sum_{\omega\in\Omega_j} p^\omega(\sum_{r\in\mathcal{R}} \beta_r^\omega A_{ijr})$ implied by the resource utilization of this allocation. In this case, the master problem is equivalent to the following stochastic matching problem that minimizes the total expected cost of the allocation patterns :

$$\min \sum_{i\in\mathcal{S}_j} [c_{ij} - \sum_{\omega\in\Omega_j} p^\omega (\sum_{r\in\mathcal{R}} \beta_r^\omega A_{ijr})]\ x_{ij} \tag{7.19}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1 \tag{7.20}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{7.21}$$

It is challenging to change an iterative method to a one-iteration procedure. The classical L-shaped procedure computes an initial solution of the master problem without considering any cuts and thus any stochastic information ; it will iteratively refine this solution by adding new cuts. The one-iteration approach waits for optimality cuts to make a better decision based on insight into forthcoming requests. We consequently set an initial solution $\overline{x}_{ij} = 0$ that allows us to delay the allocation decision of the jth request, as in Van Hentenryck and Bent (2009).

Furthermore, the optimality cuts (7.17), which are generated by the relaxed subproblems for the solution $\overline{x}_{ij}$, are the best approximation of the recourse function $Q(x_{ij}, \omega)$ at the point $\overline{x}_{ij}$. However, they undervalue $Q(x_{ij}, \omega)$ for all the other points. Since they are computed only once, the solution $\overline{x}_{ij}$ is disadvantaged in the stochastic matching problem. Initializing $\overline{x}_{ij} = 0$ gives fair optimality cuts for all the nonzero solutions.

The relaxed subproblems must now deal with this initialization. This leads to two transformations : each scenario must include the jth request (index $l = 0$ in the scenario of future requests), and we add the constraint $\sum_{i\in\mathcal{S}_j} y_{ij}^\omega = 1 - \overline{x}_{ij} = 1$ (this constraint adds only a constant part to the objective (7.19)). In each scenario, the relaxed subproblem makes now the best allocation decision for the jth request.

To summarize, the one-iteration L-shaped procedure is a descent algorithm where the direction is equal to $c_{ij} - \sum_{\omega\in\Omega_j} p^\omega(\sum_{r\in\mathcal{R}} \beta_r^\omega A_{ijr})$. In a stochastic world, this direction indicates the region of the search space where we expect to find a better solution than the current one, $\overline{x}_{ij} = 0$. However, for some constraints we may not have enough dual information to guide the algorithm to an optimal solution. We use primal information on these constraints to remove decisions from the search space via feasibility cuts.

7.4.2 Probabilistic feasibility cuts

Usually, feasibility and optimality of the current decision $\overline{x}_{ij}$ are checked for each relaxed subproblem at each iteration of the classical L-shaped procedure. In our case, since each relaxed subproblem is solved once, feasibility and optimality are not fully checked. Instead, the algorithm retrieves primal information from the solution of the relaxed subproblems.

Let $\mathcal{D}$ be the set of resources for which the corresponding constraint does not provide enough dual information to the stochastic matching problem. When solving the relaxed subproblems, we store the optimal solution $\overline{y}^{\omega}_{il}$, which gives the optimal load on the resources. We assume that the optimal load of any resource in $\mathcal{D}$ remains close to optimal for any feasible decision $\overline{x}_{ij}$. We add the feasibility cuts $\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \leq b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}^{\omega}_{il}$ to the stochastic matching problem for each scenario ω and for each resource r in $\mathcal{D}$. The jth request, which corresponds to the index $l = 0$ in each scenario, is not taken into account in the optimal load because the variables x_{ij} determine the allocation of this request.

$$\min \sum_{i\in\mathcal{S}_j}[c_{ij} - \sum_{\omega\in\Omega_j} p^{\omega}(\sum_{r\in\mathcal{R}}\beta^{\omega}_r A_{ijr})]\; x_{ij} \tag{7.22}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1 \tag{7.23}$$

$$\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \leq b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}^{\omega}_{il}, \qquad \forall r\in\mathcal{D}, \forall\omega\in\Omega_j \tag{7.24}$$

$$x_{ij}\in\{0,1\}, \qquad \forall i\in\mathcal{S}_j \tag{7.25}$$

We transform the feasibility cuts (7.24) in the spirit of chance constraints to allow flexibility : since these cuts are approximations, they may lead to an allocation decision that is overly conservative. The goal is now to respect these cuts according to a certain confidence level η, which we call the primal-ratio. These cuts now become $\mathbb{P}_{\Omega_j}[\sum_{i\in\mathcal{S}_j} A_{ijr}x_{ij} \leq b_r - \sum_{l=1}^{T^\omega}\sum_{i\in\mathcal{S}_l} A_{ilr}\overline{y}^{\omega}_{il}] \geq \eta$. Since the sample set Ω_j is finite, let Ω^r_j be a subset of scenarios for resource r such that $\sum_{\omega\in\Omega^r_j} p^{\omega} \geq \eta$. This subset is problem-related and easy to find, as we will show when we discuss our applications.

$$\min \sum_{i\in\mathcal{S}_j}[c_{ij}-\sum_{\omega\in\Omega_j}p^{\omega}(\sum_{r\in\mathcal{R}}\beta_r^{\omega}A_{ijr})]x_{ij} \tag{7.26}$$

subject to

$$\sum_{i\in\mathcal{S}_j}x_{ij}=1 \tag{7.27}$$

$$\sum_{i\in\mathcal{S}_j}A_{ijr}x_{ij}\leq b_r-\sum_{l=1}^{T^{\omega}}\sum_{i\in\mathcal{S}_l}A_{ilr}\overline{y}_{il}^{\omega}, \qquad \forall r\in\mathcal{D},\forall\omega\in\Omega_j^r \tag{7.28}$$

$$x_{ij}\in\{0,1\}, \qquad \forall i\in\mathcal{S}_j \tag{7.29}$$

This restricted stochastic matching problem takes into account the dynamic parts of the problem : the dual variables β^{ω} measure the expected cost of the resource utilization and, depending on the primal-ratio, constraints (7.28) forbid allocation decisions that might lead at the end of the horizon to infeasible or nonoptimal final solutions. The primal-ratio should thus be as small as possible to avoid the removal of optimal decisions from the search space.

7.4.3 Global online stochastic algorithm

Figure 7.1 presents the OS algorithm. This algorithm communicates with all the other parts of the system through an information system : it retrieves the state of the system before any decision and subsequently relays its decision to the rest of the system.

For each new request, the algorithm first finds feasible allocation patterns and then makes the best feasible decision. For some requests, the algorithm can save time by making decisions with an online policy instead of reoptimizing. Determining when this situation occurs is problem-dependent.

We simulate the other parts of the system in the two applications to evaluate our results in a realistic environment.

7.5 Application I : Appointment booking and scheduling problem

The online appointment booking problem (Gupta and Denton, 2008) involves finding the best appointment for each new patient as he/she arrives. The main challenge is to maintain just enough free slots for high-priority patients that may arrive in the future. Online radiotherapy appointment booking is a relatively recent application. When patients arrive

Figure 7.1 – Flow chart of global online stochastic algorithm.

in a cancer treatment facility, they must undergo a series of examinations before receiving treatment on a linac, which irradiates the malignant tumor to kill the infected cells. After the first consultation, the patients undergo a scan to locate the tumor, and then the cancer treatment is prepared by the dosimetrists. The dosimetry primarily involves planning the shape, intensity, and direction of the beams of the linac. These steps form the pretreatment phase.

The total waiting time is measured as the number of days between the first consultation and the beginning of the treatment. The management must report detailed statistics to the state authorities on a regular basis. Furthermore, the center treats palliative (i.e., high priority) patients to relieve their pain and curative (i.e., low priority) patients to maximize their chances of recovery. The operator should balance resources between these two types of patients while respecting, as far as possible, the waiting-time targets.

Two classical block scheduling heuristics have been proposed for online appointment booking problems (Petrovic et al., 2006) : "just in time" (JIT) and "as soon as possible" (ASAP). The first schedules patients on their due dates, and the second schedules them on their release dates. Petrovic et al. (2006) combine the two heuristics : JIT for the high-priority patients and ASAP for the others. Klassen and Rohleder (2004) propose general heuristics for outpatient clinics. They compare different rules such as "First-call, first-appointment" and "Low variance clients at the beginning of the schedule." They show that the latter rule is the best for several objectives, such as the client waiting time. Patrick et al. (2008a) propose a general approximate dynamic program for outpatient clinics. Sauré et al. (2012) have successfully applied this technique to a radiotherapy center. Legrain et al. (2014) solve the booking problem with an online clairvoyant algorithm.

However, none of these methods can deal with the appointment booking and scheduling problem. This is because if the pretreatment planning is not completed on time, the treatment may be postponed, and this may cause the cancellation of the linac appointment. The management aims to reduce pretreatment processing times and to avoid unnecessary linac cancellations.

We apply the OS algorithm presented in Figure 7.1 to the booking and scheduling of a radiotherapy center in Quebec, Canada. In Quebec, patients with cancer can wait a considerable time for treatment. The Quebec authorities have defined a target maximum delay of 28 days. However, the Quebec College of Physicians advises more specific targets. Palliative patients should start treatment less than three days after admission, whereas curative patients can wait 14 or 28 days depending on the type of cancer. Two tasks must be performed by two different dosimetrists before treatment can begin : the first is the preparation of the treatment and the second is its verification. Other minor tasks must be performed to complete the pretreatment ; they are modeled through a set of fixed delays. The scheduling of the pretreatment can thus be viewed as a hybrid flow-shop with recirculation (Ruiz and Vázquez-Rodríguez, 2010).

This work has been realized in collaboration with the Centre Intégré de Cancérologie de Laval (CICL). We have published an extended abstract (Legrain et al., 2015) on this application.

7.5.1 Online stochastic formulation

We now present a stochastic optimization model for the appointment booking and scheduling problem. The planning of the dosimetry and the linac appointments are represented by columns. On the arrival of patient j, the model infers the average cost of linac plans for a finite set Ω_j of future patient ($\mathcal{P}^\omega$) scenarios ω of probability p^ω. Each future patient set also contains the current patient j. Let $\mathcal{H}$ be the index set of the working days over the planning horizon, $\mathcal{B}$ the index set of Mondays, and $\mathcal{M}$ the set of available linacs. Let $\mathcal{S}_j$ be the index set of feasible linac appointment patterns for patient j, a^m_{ijk} the description of pattern $i \in \mathcal{S}_j$ ($= 1$ if the patient is treated on linac m on day k, and 0 otherwise), b_{ij} the day of the first treatment session in pattern $i \in \mathcal{S}_j$, and c_{ij} the cost of pattern $i \in \mathcal{S}_j$. This cost is a nonlinear combination of waiting times and deadline-violation penalties. The parameter r_{il} represents the end of the pretreatment for patient l in dosimetry planning pattern $i \in \mathcal{S}^D$. Let F^m_k be the number of available slots on linac m on day k, O_{day} the maximum daily number of overtime slots on linac m, O_{week} the maximum weekly number of overtime slots on linac m, and c^o the cost of an overtime slot. The variable x_{ij} is 1 if linac appointment pattern $i \in \mathcal{S}_j$ is allocated to new patient j and 0 otherwise; y^ω_{il} is 1 if linac appointment pattern $i \in \mathcal{S}_l$ is chosen for patient l in scenario $\omega \in \Omega_j$ and 0 otherwise; and v^ω_i is 1 if dosimetry planning pattern $i \in \mathcal{S}^D$ is chosen for all patients of scenario $\omega \in \Omega_j$ and 0 otherwise. Finally, z_{mk} is the number of overtime slots on linac m on day k.

$$\min \sum_{i\in\mathcal{S}_j} c_{ij}x_{ij} + \sum_{\omega\in\Omega_j} p^\omega [\sum_{l\in\mathcal{P}^\omega}\sum_{i\in\mathcal{S}_l} c_{il}y^\omega_{il} + \sum_{k\in\mathcal{H}}\sum_{m\in\mathcal{M}} c^o z^\omega_{mk}] \tag{7.30}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} + \sum_{i\in\mathcal{S}_j} y^\omega_{ij} = 1, \qquad \forall \omega\in\Omega_j \tag{7.31}$$

$$\sum_{i\in\mathcal{S}_l} y^\omega_{il} = 1, \qquad \forall \omega\in\Omega_j, \forall l\in\mathcal{P}^\omega\backslash\{j\} \tag{7.32}$$

$$\sum_{i\in\mathcal{S}^D} v^\omega_i = 1, \qquad \forall \omega\in\Omega_j \tag{7.33}$$

$$\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} + \sum_{i\in\mathcal{S}_j} b_{ij}y^\omega_{ij} - \sum_{i\in\mathcal{S}^D} r_{ij}v^\omega_i \geq 0, \qquad \forall \omega\in\Omega_j \tag{7.34}$$

$$\sum_{i\in\mathcal{S}_l} b_{il}y^\omega_{il} - \sum_{i\in\mathcal{S}^D} r_{il}v^\omega_i \geq 0, \qquad \forall \omega\in\Omega_j, \forall l\in\mathcal{P}^\omega\backslash\{j\} \tag{7.35}$$

$$\sum_{i\in\mathcal{S}_j} a^m_{ijk}x_{ij} + \sum_{l\in\mathcal{P}^\omega}\sum_{i\in\mathcal{S}_l} a^m_{ilk}y^\omega_{il} \leq F^m_k + z^\omega_{mk}, \qquad \forall m\in\mathcal{M}, \forall k\in\mathcal{H}, \forall\omega\in\Omega_j \tag{7.36}$$

$$\mathbb{1}_{\mathcal{P}_p}(j)\sum_{i\in\mathcal{S}_j} a^m_{ijk}x_{ij} + \sum_{l\in\mathcal{P}^\omega_p}\sum_{i\in\mathcal{S}_l} a^m_{ilk}y^\omega_{il} \geq z^\omega_{mk}, \qquad \forall m\in\mathcal{M}, \forall k\in\mathcal{H}, \forall\omega\in\Omega_j \tag{7.37}$$

$$\sum_{k=b}^{b+4} z^{\omega}_{mk} \leq O_{week}, \qquad \forall m \in \mathcal{M}, \forall b \in \mathcal{B}, \forall \omega \in \Omega_j \tag{7.38}$$

$$z^{\omega}_{mk} \leq O_{day}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \tag{7.39}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j \tag{7.40}$$

$$y^{\omega}_{il} \in \{0,1\}, \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in \mathcal{S}_l, \forall \omega \in \Omega_j \tag{7.41}$$

$$v^{\omega}_{i} \in \{0,1\}, \qquad \forall i \in \mathcal{S}^D, \forall \omega \in \Omega_j \tag{7.42}$$

$$z^{\omega}_{mk} \geq 0, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H}, \forall \omega \in \Omega_j \tag{7.43}$$

Constraints (7.31) and (7.32) ensure respectively that patient j and all future patients are scheduled on linacs. Constraints (7.33) choose a dosimetry schedule for each scenario ω. The columns x_{ij}, y^{ω}_{il} representing patient appointments on linacs are all inserted once at the beginning because the sets $\mathcal{S}_j$ and $\mathcal{S}_l$ are small. However, the columns v^{ω}_{i} are generated during the solution process because the set $\mathcal{S}^D$ is large. A genetic algorithm, presented in Appendix 7.8.1, is used for the column generation procedure. Constraints (7.34) and (7.35) are precedence constraints : they ensure respectively that patient j and all future patients have completed their pretreatment in time for their first linac treatment. Constraints (7.36) verify that the capacity (including the overtime) of each linac is not exceeded. Constraints (7.37) ensure that only palliative patients are scheduled in overtime slots ($\mathcal{P}^{\omega}_{p}$ is a subset of $\mathcal{P}^{\omega}$ containing the palliative patients). Constraints (7.38) and (7.39) bound the weekly and daily overtime on each linac. Constraints (7.40), (7.41), (7.42), and (7.43) are domain constraints. Finally, the objective (7.30) is divided into two parts : the cost of the plan for patient j and the average future cost of the linac plans.

7.5.2 Integer subproblems

The OS formulation is transformed as shown in the methodology presented in Section 7.4. The integer subproblems solve the booking and scheduling problem for a solution $\overline{x}_{ij}$ and for each scenario ω.

$$Q(\overline{x}_{ij}, \omega) \;=\; \min \sum_{l \in \mathcal{P}^{\omega}} \sum_{i \in \mathcal{S}_l} c_{il} y^{\omega}_{il} + \sum_{k \in \mathcal{H}} \sum_{m \in \mathcal{M}} c^{o} z^{\omega}_{mk} \tag{7.44}$$

subject to

$$\sum_{i \in \mathcal{S}_j} y^{\omega}_{ij} = 1 - \sum_{i \in \mathcal{S}_j} \overline{x}_{ij} \tag{7.45}$$

$$\sum_{i\in\mathcal{S}_l} y_{il}^{\omega} = 1, \qquad \forall l \in \mathcal{P}^{\omega}\backslash\{j\} \tag{7.46}$$

$$\sum_{i\in\mathcal{S}^D} v_i^{\omega} = 1, \tag{7.47}$$

$$\sum_{i\in\mathcal{S}_l} b_{ij} y_{ij}^{\omega} - \sum_{i\in\mathcal{S}^D} r_{ij} v_i^{\omega} \geq -\sum_{i\in\mathcal{S}_j} b_{ij}\overline{x}_{ij}, \tag{7.48}$$

$$\sum_{i\in\mathcal{S}_l} b_{il} y_{il}^{\omega} - \sum_{i\in\mathcal{S}^D} r_{il} v_i^{\omega} \geq 0, \qquad \forall \omega \in \Omega_j, \forall l \in \mathcal{P}^{\omega}\backslash\{j\} \tag{7.49}$$

$$\sum_{l\in\mathcal{P}^{\omega}} \sum_{i\in\mathcal{S}_l} a_{ilk}^m y_{il}^{\omega} \leq F_k^m + z_{mk}^{\omega} - \sum_{i\in\mathcal{S}_j} a_{ijk}^m \overline{x}_{ij}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{7.50}$$

$$\sum_{l\in\mathcal{P}_p^{\omega}} \sum_{i\in\mathcal{S}_l} a_{ilk}^m y_{il}^{\omega} \geq z_{mk}^{\omega} - \mathbb{1}_{\mathcal{P}_p}(j) \sum_{i\in\mathcal{S}_j} a_{ijk}^m \overline{x}_{ij}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{7.51}$$

$$\sum_{k=b}^{b+4} z_{mk}^{\omega} \leq O_{week}, \qquad \forall m \in \mathcal{M}, \forall b \in \mathcal{B} \tag{7.52}$$

$$z_{mk}^{\omega} \leq O_{day}, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{7.53}$$

$$y_{il}^{\omega} \in [0,1], \qquad \forall l \in \mathcal{P}^{\omega}, \forall i \in \mathcal{S}_l \tag{7.54}$$

$$v_i^{\omega} \in [0,1], \qquad \forall i \in \mathcal{S}^D \tag{7.55}$$

$$z_{mk}^{\omega} \geq 0, \qquad \forall m \in \mathcal{M}, \forall k \in \mathcal{H} \tag{7.56}$$

7.5.3 Probabilistic feasibility cuts

Constraint (7.48) can be infeasible if new patient j is scheduled too early on the linacs. Let $\mathcal{D}$ be the singleton formed by this constraint. For each scenario ω, if $\overline{v}^{\omega}$ is the solution of the relaxed subproblems, we add the feasibility cuts $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \sum_{i\in\mathcal{S}^D} r_{ij}\overline{v}_i^{\omega}$. The sum $\sum_{i\in\mathcal{S}^D} r_{ij}\overline{v}_i^{\omega} = \overline{r}_j^{\omega}$ is now a constant and represents the earliest starting time for a linac appointment for patient j in scenario ω. Furthermore, if $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega_0}$ holds for a scenario ω_0, $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega}$ will also hold for any scenario ω such that $\overline{r}_j^{\omega} \leq \overline{r}_j^{\omega_0}$. Consequently, adding just one constraint suffices to represent all the constraints associated with each scenario.

For a primal-ratio η, we thus choose the scenario $\omega_0 = \text{argmin}_{\omega\in\Omega_j}\{\overline{r}_j^{\omega} \,\big|\, |\{\omega_1 \in \Omega_j | \overline{r}_j^{\omega_1} \leq \overline{r}_j^{\omega}\}| \geq \eta\,|\Omega_j|\}$ (where $|\ .\ |$ indicates cardinality). If $\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega_0}$ holds, $\mathbb{P}_{\Omega_j}\{\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}_j^{\omega}\} \geq \eta$. The subset Ω_j^0 is thus equal to the singleton $\{\omega_0\}$ for the only constraint in $\mathcal{D}$.

7.5.4 Restricted stochastic matching problem

Let δ^{ω}, β_{mk}^{ω}, and γ_{mk}^{ω} be the dual variables associated respectively with the constraints (7.48), (7.50), and (7.51). The stochastic dual cost associated with pattern i of patient j is defined to be $\sum_{\omega\in\Omega_j} p^{\omega}[\delta^{\omega} b_{ij} + \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{H}}(\beta_{mk}^{\omega} + \mathbb{1}_{\mathcal{P}_p}(j)\gamma_{mk}^{\omega})a_{ijk}^m]$. Therefore, the restricted stochastic matching problem is

$$Z^* = \min \sum_{i\in\mathcal{S}_j} \{c_{ij} - \sum_{\omega\in\Omega_j} p^\omega[\delta^\omega b_{ij} + \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{H}} (\beta^\omega_{mk} + \mathbb{1}_{\mathcal{P}_p}(j)\gamma^\omega_{mk})a^m_{ijk}]\}x_{ij}$$

subject to

$$\sum_{i\in\mathcal{S}_j} x_{ij} = 1$$
$$\sum_{i\in\mathcal{S}_j} b_{ij}x_{ij} \geq \overline{r}^\omega_j, \qquad \forall\omega \in \Omega^0_j$$
$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{S}_j$$

The appointment patterns for the linacs in set $\mathcal{S}_j$ are all feasible and thus respect constraints (7.36)–(7.39) for patient j. The restricted stochastic matching problem determines which feasible pattern i, with a starting time greater than $\overline{r}^{\omega_0}_j$, has the minimum expected cost Z^*.

7.5.5 Global online stochastic algorithm

Figure 7.2 illustrates the steps of the OS algorithm for this application. The information system must communicate the states of the linacs and the dosimetrists. The simulator generates new requests based on a probability distribution and schedules daily pending dosimetry tasks using a constraint program, presented in Appendix 7.8.2. This program also checks the feasibility of an allocation pattern during the filtering.

The online algorithm uses an online policy to book palliative patients. Since they have a high priority, the ASAP heuristic gives good results and avoids the need to solve the relaxed subproblems.

7.5.6 Experiments

All the experiments were run over 8 threads on a computer with an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz and 32 GB of memory. We used CPLEX and CP OPTIMIZER 12.6.

We first study the behavior of the algorithm and then present results for a real data set. The scenarios used for the online algorithms were drawn from the empirical distribution of the CICL. A large proportion (70%) of the curative patients are known in advance because they

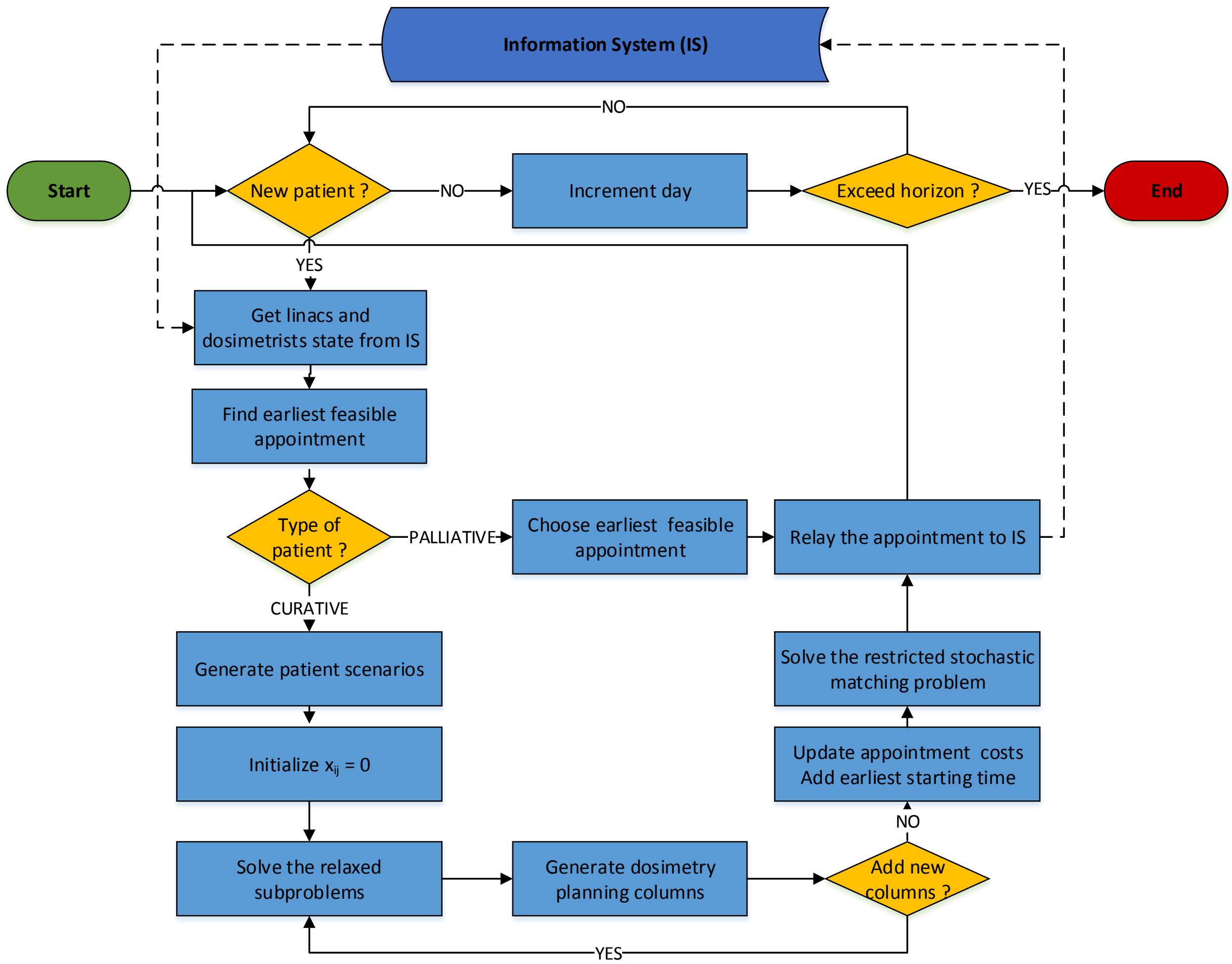


Figure 7.2 – Flow chart of global CICL online stochastic algorithm.

have already undergone surgery and/or chemotherapy in the center.

Sensitivity analysis

There are two linacs with 29 slots plus 3 overtime slots and 4 dosimetrists. Three statistics computed over 30 runs are used to analyze the algorithm presented in Figure 7.2 :

— First appointment canceled : the number of patients for which the first treatment session is canceled because the pretreatment is late. This corresponds to the number of violated precedence constraints (7.34).

— Overdue : the number of patients for which the waiting-time target has not been met.

— Objective : the sum over all patients of deadline violations and waiting-time penalties.

The primal-ratio η can take into account the risk aversion of the operator in relation to the precedence constraints. Figure 7.3 shows the evolution of the three statistics as a function of the primal-ratio.

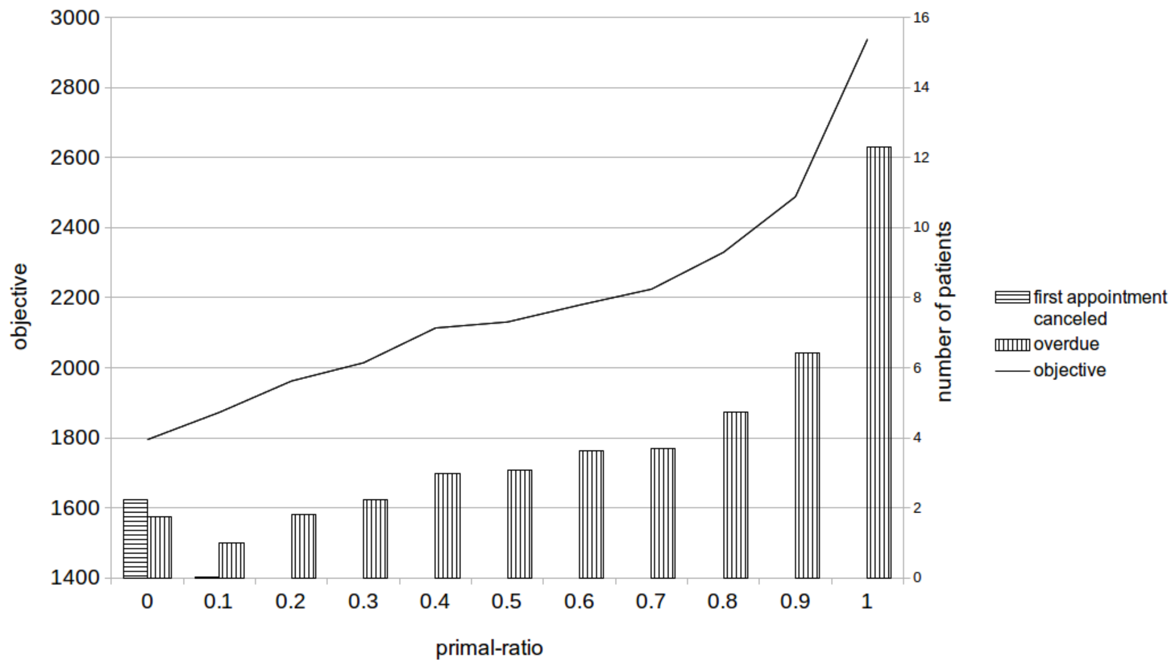


Figure 7.3 – Analysis of the value of the primal-ratio.

The left axis gives the average value of the objective, and the right axis counts the average number of patients in each of the groups. As the primal-ratio increases, the objective value grows. Indeed, the more careful the operator, the later the patients will be given their first linac appointment. When the primal-ratio is high, the algorithm tends to delay curative patients to avoid canceling treatment sessions because of pretreatment delays. The objective function is also high because waiting time and deadline violations are penalized. In contrast, when the primal-ratio is low, the objective function is also low because the patients have short waiting times ; however, some linac appointments will be canceled. Since this situation must be avoided, we set the primal-ratio to $\eta = 0.2$.

Results

We evaluate the approach on a large real instance from the CICL with 1529 patients over 248 working days. The center operates with 4 dosimetrists and 4 linacs. Table 7.1 compares three algorithms according to various criteria : 1) the number of violated precedence constraints (7.34) (i.e., first appointment canceled), 2) the number of targets not met (i.e., overdue), 3) the average waiting time, and 4) the number of overtime slots used. The CICL algorithm is the greedy procedure that is currently used at the facility. The online clairvoyant algorithm is the the procedure presented in (Legrain et al., 2014). These two algorithms, unlike the third, do not take into account the dosimetry planning and assume a fixed delay. Finally, the OS algorithm follows the procedure presented in Figure 7.2 and uses 8 scenarios (because the computer can run 8 threads simultaneously).

Table 7.1 – Comparison of algorithms on a real instance.

Algorithm	Appointment canceled	Target not met			Average waiting time			Overtime slots
		>3 days	>14 days	>28 days	3 days	14 days	28 days	
CICL	230	373 (25)	104	0	3.45	12.58	12.63	111
Online Clairvoyant	107	335 (25)	67 (1)	0	3.27	12.23	14.77	19
Online Stochastic	1	326 (24)	119 (5)	0	3.23	14.04	18.43	8

Table 7.1 shows that the OS algorithm outperforms the two other procedures. It has just one canceled appointment whereas the CICL procedure has 230 cancellations and the online clairvoyant has 107. Furthermore, the number of targets not met remains stable. The OS algorithm delays the curative patients to increase the likelihood that the pretreatment will be completed on time. The curative patients do not do as well on criteria 2) and 3), but the palliative patients do better on these criteria. Finally, the computational time for the OS algorithm is an average of 20 seconds per patient.

7.6 Application II : Task assignment and routing problem

The task assignment and routing problem arises in the management of warehouse operations (Gu et al., 2007). Orders arrive in the warehouse and are treated by the manager or the warehouse management system (WMS), which splits or groups them into pick-up tasks. Additional tasks, such as put-away, counting, and replenishing must also be performed. A route is composed of a starting location, various intermediate stops, and a final destination. Whenever a worker completes a task, the WMS must dynamically choose the next task for that worker.

In general, the tasks are given priorities, and the highest-priority tasks should be completed first. However, a few low-priority tasks may be interleaved to avoid unnecessary deadheads (i.e., trips without any goods). For example, a transition task may be inserted between the final destination of the previous task and the starting location of the next, resulting in more efficient routes.

The offline task assignment problem has been widely studied (Ernst et al., 2004). Corominas et al. (2006) propose a heuristic to assign tasks in the service industry once the shift schedule has been fixed. Bard and Wan (2006) develop a tabu search procedure for a similar problem. Their system has been tested on real data from a U.S. Postal Service mail processing and distribution center. Boyer et al. (2014) propose a grammar-based approach to include the task assignment problem at the shift scheduling level.

However, few studies investigate the online task assignment problem. For a warehouse, Rubrico et al. (2011) reschedule the current routes each time a new task arrives. They develop VRP heuristics to compute new routes as well as heuristics to adapt the current routes. These heuristics give good results when there is a balanced mix of static and dynamic tasks. However, they are pure online algorithms : no stochastic information about future tasks is considered.

In this application, which has been realized in collaboration with JDA Software, we investigate the task assignment and routing problem in an OS fashion. The challenge in deciding the next task for each worker lies in finding a balance between minimizing deadheads and performing urgent tasks quickly enough. We choose to explicitly maximize the sum of the priorities of the completed tasks over the horizon, while the deadheads will be implicitly minimized through the generation of patterns.

7.6.1 Online stochastic formulation

In this application, the requests normally represent the tasks. However, an assignment becomes effective only when the employee has completed his/her previous nonpreemptive task. To avoid unnecessary computation, we can postpone the allocation until an employee is available. We define the requests as a sequence of employees that have completed an assignment (an employee will normally appear several times in this sequence). Consequently, there is a set $\mathcal{T}$ of waiting tasks, which might not be empty at the beginning of the horizon.

We present a stochastic optimization model for this task assignment and routing problem. When the jth request arrives, i.e., employee $r \in \mathcal{R}$ has finished an assignment, the WMS assigns a new task to this employee, and thus it dynamically builds a global set of efficient routes for all the workers. The model infers the average cost of completing the current routes for a finite set Ω_j of future task scenarios. Since solving a stochastic vehicle routing problem is computationally demanding, the algorithm solves this problem once for all the employees and stores their future assignments. When the jth request arrives, the algorithm either retrieves a stored assignment or solves the OS problem if the previously computed assignment has already been performed.

The scenario $\omega \in \Omega_j$, with probability p^ω, represents the set $\mathcal{T}^\omega$ of future tasks, which contains the set $\mathcal{T}$ (common to all the scenarios) of waiting tasks. Variable x_{ir} is 1 if waiting task $i \in \mathcal{T}$ with priority c_i is allocated to employee r, and 0 otherwise. Each new task must form the beginning of a future route $p \in \mathcal{S}_r$. This route is described by a pattern for employee r : parameter a_{irp} (b_{irp}) is 1 if task i is on route p (starts route p), and 0 otherwise. The variable y_i^ω is 1 if task $i \in \mathcal{T}^\omega$ is completed in scenario ω, and 0 otherwise ; v_{rp}^ω is 1 if route p is allocated to employee r, and 0 otherwise.

$$\max \sum_{i \in \mathcal{T}} \sum_{r \in \mathcal{R}} c_i x_{ir} + \sum_{\omega \in \Omega_j} p^\omega [\sum_{i \in \mathcal{T}^\omega} c_i y_i^\omega] \tag{7.57}$$

subject to

$$\sum_{r \in \mathcal{R}} x_{ir} + y_i^\omega \leq 1, \qquad \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j \tag{7.58}$$

$$\sum_{p \in \mathcal{S}_r} v_{rp}^\omega \leq 1, \qquad \forall r \in \mathcal{R},\ \forall \omega \in \Omega_j \tag{7.59}$$

$$\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{S}_r^\omega} a_{irp} v_{rp}^\omega \geq y_i^\omega, \qquad \forall i \in \mathcal{T}^\omega,\ \forall \omega \in \Omega_j \tag{7.60}$$

$$\sum_{p \in \mathcal{S}_r^\omega} b_{irp}^\omega v_{rp}^\omega \geq x_{ir}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j \tag{7.61}$$

$$x_{ir} \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{7.62}$$

$$y_i^\omega,\ v_{rp}^\omega \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T}^\omega,\ \forall \omega \in \Omega_j,\ \forall p \in \mathcal{S}_r^\omega \tag{7.63}$$

The objective (7.57) maximizes the sum of the priorities of the current and expected assignments. Constraints (7.58) verify that a potential next task, that must be in the set $\mathcal{T}$ of waiting tasks, is completed at most once in each scenario. Constraints (7.59) ensure that at most one route is assigned to each worker. The columns v_{rp}^ω are generated during the solution process because the sets $\mathcal{S}_r$ are large (see Appendix 7.8.1 for the column generation procedure). Constraints (7.60) link each route to the included tasks. Constraints (7.61) ensure that the next assignment of each employee starts a feasible route. Finally, Constraints (7.62) and (7.63) define x_{ir}, y_i^ω, and v_{rp}^ω as binary variables.

7.6.2 Integer subproblems

The OS formulation is transformed as shown in the methodology presented in Section 7.4. The integer subproblems solve a VRP for a solution $\overline{x}_{ir}$ and for each scenario ω.

$$Q(\overline{x}_{ir}, \omega) = \max \sum_{i \in \mathcal{T}^\omega} c_i y_i^\omega \tag{7.64}$$

subject to

$$y_i^\omega \leq 1 - \sum_{r \in \mathcal{R}} \overline{x}_{ir}, \qquad \forall i \in \mathcal{T} \tag{7.65}$$

$$\sum_{p \in \mathcal{S}_r} v_{rp}^\omega \leq 1, \qquad \forall r \in \mathcal{R} \tag{7.66}$$

$$\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{S}_r^\omega} a_{irp} v_{rp}^\omega \geq y_i^\omega, \qquad \forall i \in \mathcal{T}^\omega \tag{7.67}$$

$$\sum_{p \in \mathcal{S}_r^\omega} b_{irp}^\omega v_{rp}^\omega \geq \overline{x}_{ir}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{7.68}$$

$$y_i^\omega,\ v_{rp}^\omega \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T}^\omega,\ \forall p \in \mathcal{S}_r^\omega \tag{7.69}$$

7.6.3 Probabilistic feasibility cuts

Let $\mathcal{D}$ be the subset of constraints (7.68). The dual variables associated with these constraints are null if $\overline{x}_{ir} = 0$, i.e., task i is not allocated to employee r. If $\overline{v}_{rp}^\omega$ is the solution of the subproblem, we add the feasibility cut $\sum_{p \in \mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \geq x_{ir}$ in the following restricted stochastic matching problem for each task i, each employee r, and each scenario ω. Since the variables x_{ij} are binary, we transform these cuts to $\lceil \sum_{p \in \mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \rceil \geq x_{ir}$. Finally, these constraints prevent the assignment of a task that is not at the beginning of an active route (i.e., $\overline{v}_{rp}^\omega > 0$) in each solution of the relaxed subproblem associated with scenario $\omega \in \Omega_j$.

For a primal-ratio η, let $\mathcal{T}_r$ be the set of the feasible next tasks that are at the beginning of at least $\eta|\Omega_j|$ active routes of employee r. If the subset $\mathcal{T}_r$ is empty, we instead define $\mathcal{T}_r = \{i \in \text{argmax}_{i_0 \in \mathcal{T}}(|\{\omega \in \Omega_j | \sum_{p \in \mathcal{S}_r^\omega} b_{i_0 rp}^\omega \overline{v}_{rp}^\omega > 0\}|)\}$, which is the set of tasks that occur the most often at the beginning of an active route. The subset Ω_j^{ir} is thus equal to the singleton $\{\omega_{ir} | \sum_{p \in \mathcal{S}_r^\omega} b_{irp}^{\omega_{ir}} > 0, \text{if } i \in \mathcal{T}_r,\ \sum_{p \in \mathcal{S}_r^\omega} b_{irp}^{\omega_{ir}} = 0, \text{otherwise}\}$ for each employee r and each task i.

7.6.4 Restricted stochastic matching problem

Let β_i^ω and δ_i^ω be the dual variables associated respectively with constraints (7.65) and (7.68). The stochastic priority associated with task i for worker r is defined to be $\sum_{\omega \in \Omega_j} p^\omega [\beta_i^\omega + \delta_i^\omega]$. Therefore, the restricted stochastic matching problem is

$$Z^* = \max \sum_{r \in \mathcal{R}} \sum_{i \in \mathcal{F}_r} (c_i - \sum_{\omega \in \Omega_j} p^\omega [\beta_i^\omega + \delta_i^\omega]) x_{ir} \tag{7.70}$$

subject to

$$\sum_{r\in\mathcal{R}} x_{ir} \leq 1, \qquad \forall i \in \mathcal{T} \tag{7.71}$$

$$\sum_{i\in\mathcal{T}} x_{ir} \leq 1, \qquad \forall r \in \mathcal{R} \tag{7.72}$$

$$x_{ir} \leq \lceil \sum_{p\in\mathcal{S}_r^\omega} b_{irp}^\omega \overline{v}_{rp}^\omega \rceil, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T},\ \forall \omega \in \Omega_j^{ir} \tag{7.73}$$

$$x_{ir} \in \{0,1\}, \qquad \forall r \in \mathcal{R},\ \forall i \in \mathcal{T} \tag{7.74}$$

Constraints (7.71) and (7.72) are added because this matching problem is solved once for all the employees. We explicitly maximize the total expected priority Z^*, which does not depend on the employee : the dual variables measure the expected reward of performing a task. We implicitly minimize the deadheads through the probabilistic feasibility cuts (7.73) : based on the primal solution $\overline{v}_{rp}^\omega$, these constraints forbid certain next assignments for each employee.

7.6.5 Global online stochastic algorithm

Figure 7.4 illustrates the steps of the OS algorithm for this application. The information system must communicate the states of the employees and the tasks waiting in the queue. The simulator generates new requests based on the employee availability and updates the employee locations in the warehouse as well as the queue.

For each new request, the algorithm either assigns the stored next task to an employee if the previous reoptimization has already computed this assignment or reoptimizes the whole routing problem.

7.6.6 Experiments

The planning horizon is 2 hours, and the workers start and finish at the rest area. All the experiments were run over 8 threads on a server with an Intel(R) Xeon(R) E5530 CPU @ 2.40 GHz and 24 GB of memory. We use CPLEX 12.5.0 and set the CPU time to two hours because of the planning horizon.

We built instances based on real data sets from one of JDA Software's clients. The scenarios were based on a random task selection from the client database. The queue contains half of the total tasks at the beginning of the horizon, as in Rubrico et al. (2011).

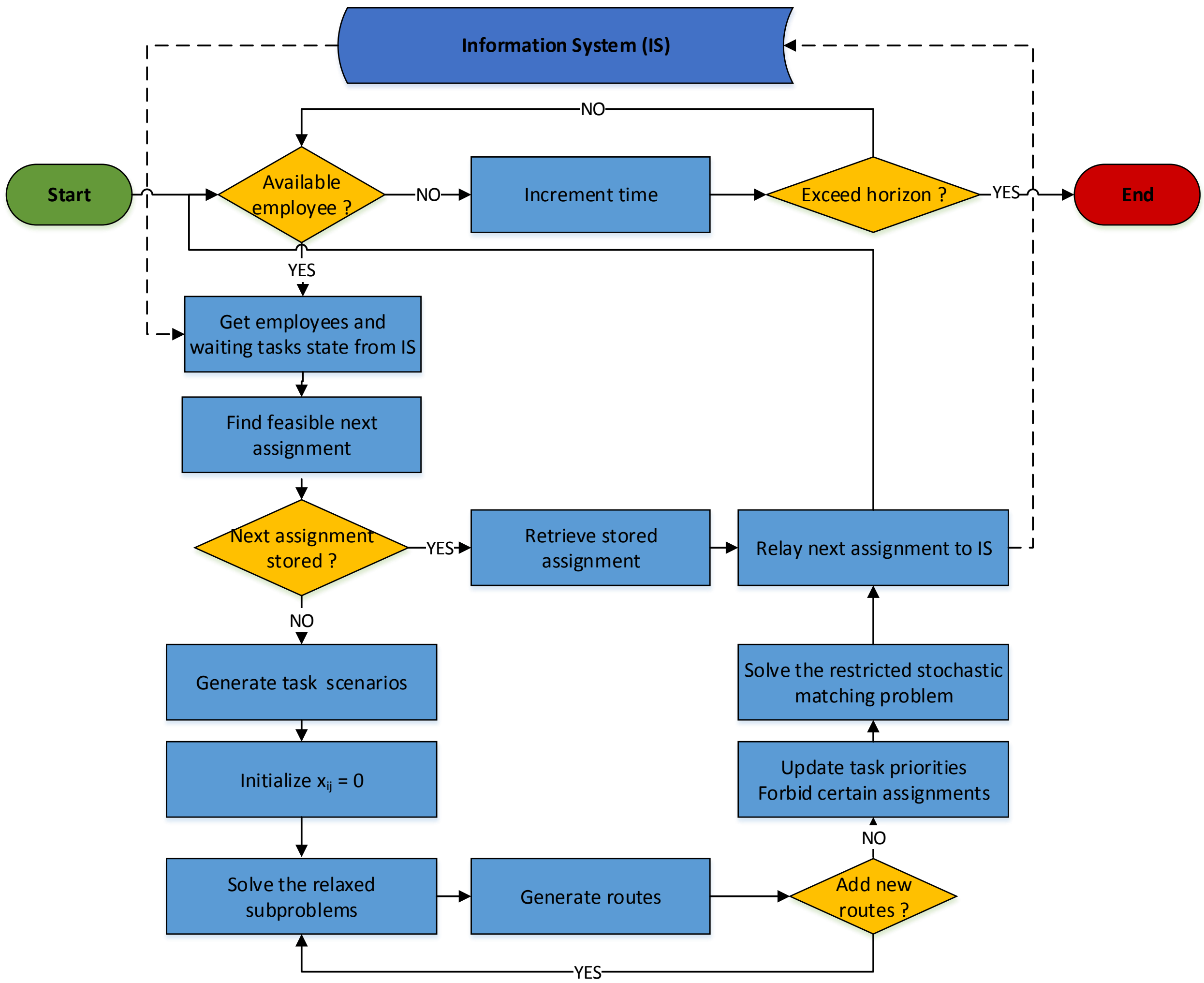


Figure 7.4 – Flow chart of global JDA online stochastic algorithm.

Sensitivity analysis

We report the results of a sensitivity analysis based on an instance with 150 tasks and 6 employees. We generate 10 scenarios, and the primal-ratio η controls the quality of the solution. Figure 7.5 shows the evolution of the average objective (computed over 30 runs) as a function of the primal-ratio.

As the primal-ratio grows, the objective value increases until $\eta = 0.5$ and then it starts to level off. This indicates that the primal solutions of the relaxed subproblems give more and more information about the efficiency of the routes. To avoid unnecessary primal noise, we set the primal-ratio to $\eta = 0.5$ for the results below.

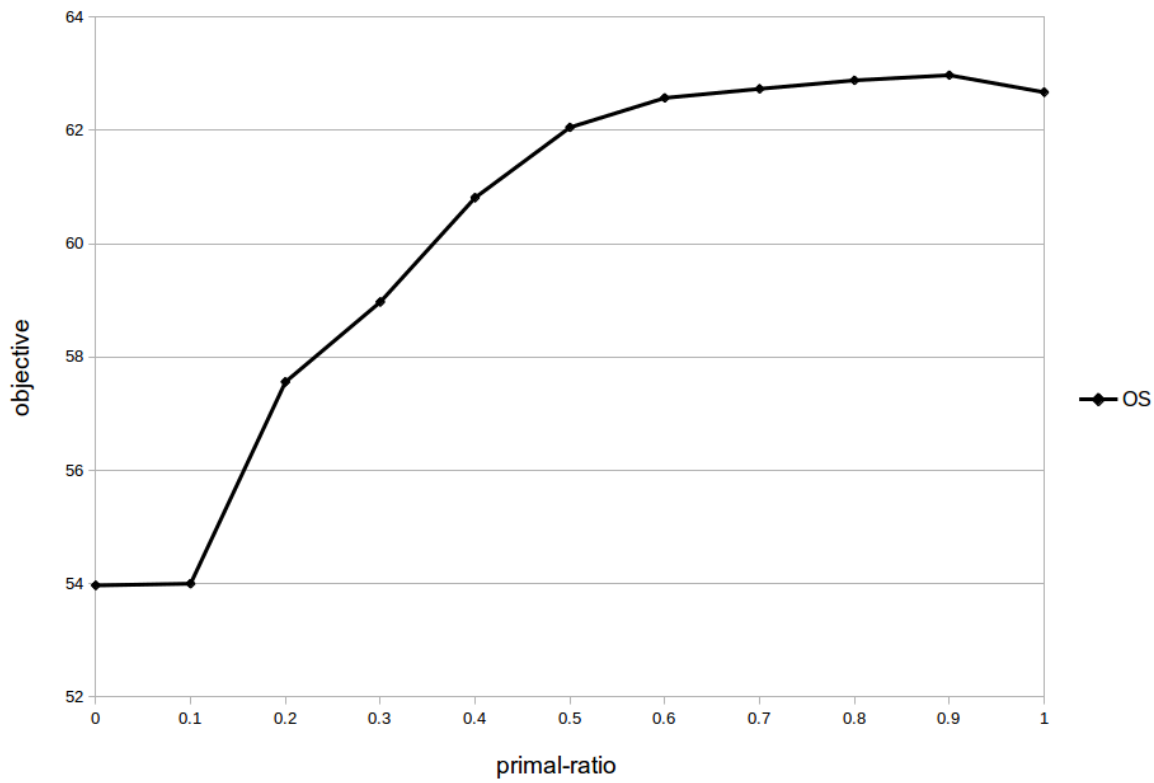


Figure 7.5 – Analysis of the value of the primal-ratio.

Results

We compare the OS algorithm to a greedy heuristic similar to that used in the WMS of JDA Software. The JDA procedure chooses the maximum priority task in the same zone as the employee. Our algorithm uses five scenarios or all the available computational time (i.e., two hours).

Preliminary tests show that the OS algorithm performs poorly (i.e., worse than the JDA procedure) without the probabilistic feasibility cuts. Indeed, the one-iteration L-shaped procedure takes into account only the expected reward of a task and thus allocates the tasks with the highest expected priorities to the employees. Deadheads are implicitly minimized in the relaxed subproblems but not in the stochastic matching problem when the feasibility cuts are absent.

The instances used for the comparison contain from 13 to 26 employees and, in average, 35 tasks per employee. Figure 7.6 plots for each instance the improvement of the objective as a function of the increase in the number of completed tasks. The improvements are measured as relative gains of the OS algorithm against the JDA procedure. The OS algorithm clearly

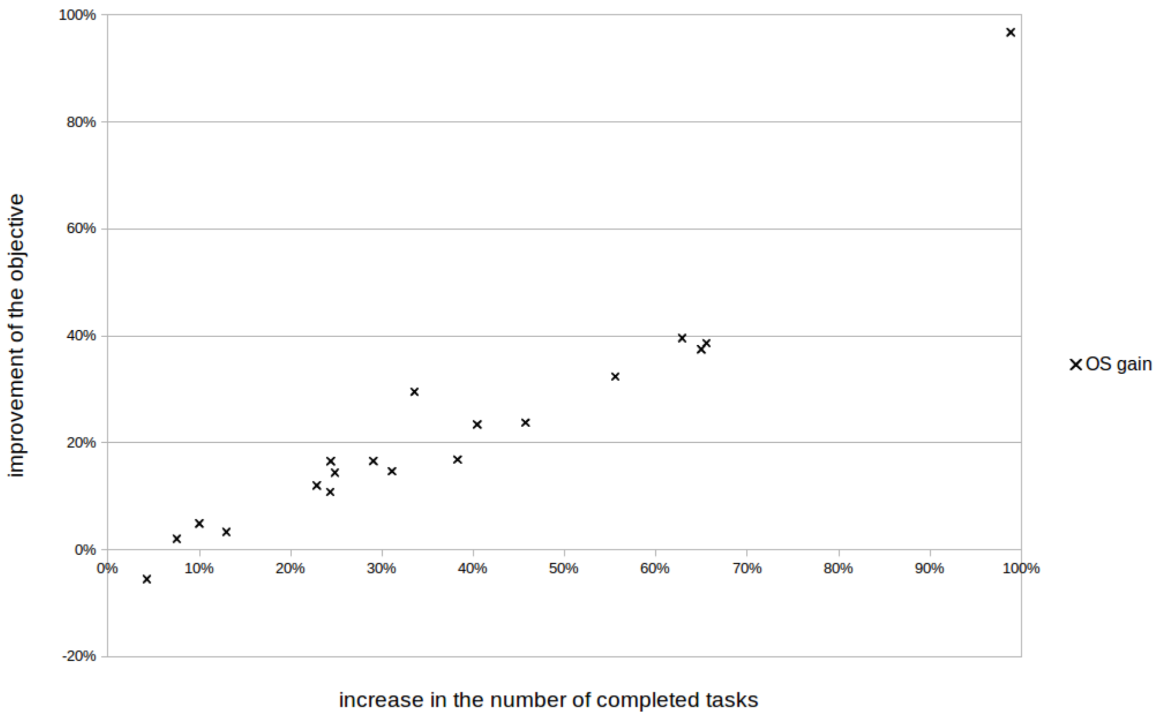


Figure 7.6 – Gain of the OS algorithm on real instances.

outperforms the JDA algorithm, as, in average, it increases of 35% the number of completed tasks and raises of 20% the objective function. The observed improvements correspond to tremendous productivity gains for a WMS.

7.7 Conclusion

In this paper, we have proposed a mathematical-programming-based framework for general OS resource allocation problems. We model the problem as a resource allocation problem via Dantzig–Wolfe decomposition and then as a two-stage program with fixed recourse using classical stochastic optimization tools. Finally, we apply Benders decomposition. We build a one-iteration L-shaped procedure to quickly obtain dual information about the future resource load. We compute the linear relaxation of an integer subproblem (an offline resource allocation problem) for each scenario to infer this load. When we have insufficient information for a resource (e.g., the dual variables associated with the resource are null), we add probabilistic feasibility cuts based on primal solutions to the master problem to remove infeasible and nonoptimal decisions.

For each new request, the online algorithm either applies an online policy or reoptimizes the resource allocation problem. At each reoptimization, we solve a restricted stochastic matching problem (i.e., the master problem) to allocate the new request according to the stochastic information about the future resource load and about expected infeasible and nonoptimal decisions. Two applications have illustrated the modeling process and demonstrated the efficiency of the proposed framework on real data sets. The OS algorithm outperforms existing procedures for both applications.

We believe that many online decision problems could be solved by our algorithm. The main challenge is to model the problem as a resource allocation problem via a Dantzig–Wolfe decomposition. One can then apply the steps of the framework.

7.8 Appendices

This section presents the specialized algorithms for generating the columns and checking the feasibility of a given allocation pattern.

7.8.1 Specialized algorithms for column generation procedure

The appointment and booking problem needs a specialized column generation procedure, while the task assignment and routing problem is a classical VRP in its offline version.

The network for the latter problem can easily be described. The tasks are nodes with a serving time and a cost (i.e., the priority). The travel time between tasks is represented by free arcs. The last routes must finish before the end of the horizon. The difficulty of this problem lies in the size of the network, which is complete. The OS algorithm solves only a linear relaxation of this problem for each scenario. The problem is decomposed using column generation (Desaulniers et al., 2005). The master problem is solved using a linear programming solver, and the subproblems, which are just resource-constrained shortest path problems in the complete network, are solved via dynamic programming.

The columns v_i^ω of the appointment and booking problem are generated during the solution process using a genetic algorithm inspired by Bertel and Billaut (2004).

The dosimetry scheduling problem has two tasks and a time window per job. These time windows correspond to fixed delays : they ensure that all the other pretreatment tasks are performed. The two tasks are the preparation and the verification of the treatment by a dosimetrist. Since the OS formulation follows a Dantzig–Wolfe scheme (Gélinas and Soumis, 2005), this flow-shop problem minimizes the weighted completion time (due to the dual variables linked to constraints (7.34) and (7.35)). Algorithm 11 thus tries to build in time T

Algorithm 11 Genetic Algorithm

Population : list of chromosomes : $P := [\,]$
Initialization : add $N/4$ chromosomes built with dispatching rules and fill the rest with random chromosomes
while solving time $< T$ AND not enough different chromosomes with a negative reduced cost **do**
  for all chromosomes c in population P **do**
    Cyclic Crossover : with probability p_c, cross c with a random different chromosome

    Mutation : with probability p_m, randomly swap two positions of c
    Intensification : with probability p_l, make a small local search with insertions on c
  end for
  Add all new chromosomes to P
  Selection : keep the best N chromosomes
end while

a population P of N chromosomes with a negative reduced cost. A chromosome is a sequence of tasks and represents a dosimetry plan. A task occurs twice in a chromosome : the first appearance corresponds to the preparation of the dosimetry, and the second corresponds to the verification. A plan is made from a chromosome by simply scheduling each task in the chromosome as early as possible. The cyclic crossover is also presented in Bertel and Billaut (2004). Finally, the intensification phase aims to improve some of the chromosomes : it tests several insertions to see if they decrease the reduced cost.

7.8.2 Specialized algorithms for checking allocation pattern feasibility

The feasibility of a pattern allocation (i.e., the next task to perform) is easily checked for the task assignment and routing problem : the algorithm checks that the employee has time to return to the rest area after performing an assignment. However, for the appointment and booking problem, the algorithm must solve a flow-shop problem to determine the feasibility of the dosimetry plan induced by an allocation pattern (i.e., a linac appointment).

The genetic algorithm quickly computes several columns, but it is impossible to know if a dosimetry plan is infeasible or optimal. To check the feasibility of an allocation pattern and to schedule the daily tasks at the dosimetry, we introduce a constraint program. It can check feasibility and find better solutions, but it takes more computational time. Let N_d be the number of dosimetrists and $\mathcal{P}_k$ be the set of patients waiting for dosimetry on day k (they correspond to jobs in this flow-shop problem). The variables C_j represent the completion time of the dosimetry for job j. The activities are interval variables in constraint programming.

They are defined by four linked variables : the beginning, the end, the length, and the presence of an interval. The variables t_{ij}, t_j^d, and t_{ij}^d are thus activities, and for job j they represent, respectively, the ith dosimetry task, the only dosimetry task performed by the dth dosimetrist, and the ith dosimetry task performed by the dth dosimetrist. Some activities are of course optional and have length zero ; the compulsory tasks are the activities t_{ij}. Finally, we aim to minimize over all the patients the square of the tardiness, $(\max(0, C_j - b_j))^2$, and the weighted completion time, $w_j C_j$. The tardiness is more important because the dosimetry plan must be feasible. However, the weighted completion time should break any ties.

$$\min \sum_{j \in \mathcal{P}_k} [(\max(0, C_j - b_j))^2 + w_j C_j] \tag{7.75}$$

subject to

$$\begin{aligned}
alternative(t_{ij}, d = 1 \dots N_d, t_{ij}^d), &\qquad \forall i = 1 \dots T, \forall j \in \mathcal{P}_k && (7.76)\\
sequential\ machine(t_{ij}^d, i = 1 \dots T, j \in \mathcal{P}_k), &\qquad \forall d = 1 \dots N_d && (7.77)\\
alternative(t_j^d, i = 1 \dots T, t_{ij}^d), &\qquad \forall d = 1 \dots N_d, \forall j \in \mathcal{P}_k && (7.78)\\
t_{ij}.end \le t_{(i+1)j}.begin, &\qquad \forall i = 1 \dots T-1, \forall j \in \mathcal{P}_k && (7.79)\\
C_j \ge t_{Tj}.end, &\qquad \forall j \in \mathcal{P}_k && (7.80)\\
t_{ij}, t_j^d, t_{ij}^d \in I_j, &\qquad \forall j \in \mathcal{P}_k && (7.81)\\
C_j \in Days, &\qquad \forall j \in \mathcal{P}_k && (7.82)
\end{aligned}$$

The alternative global constraints (7.76) ensure that only one dosimetrist performs the ith task for job j. The sequential resource constraints (7.77) ensure that each dosimetrist executes one task at a time. The global constraints (7.78) ensure that a dosimetrist completes at most one task for job j. Constraints (7.79) simply ensure that the tasks of a job are performed in the right sequence. Constraints (7.80) compute the completion time of each job j. Finally, constraints (7.81) and (7.82) describe the domains of the variables, where the set $Days$ represents the days of the planning horizon and the set I_j defines the discretized domain for each job (the possible ready and due dates are the bounds of the set).

CHAPITRE 8 DISCUSSION GÉNÉRALE

Dans cette thèse, trois problèmes d'ordonnancement en santé sont étudiés : la prise de rendez-vous en radiothérapie, la construction d'emploi du temps d'un bloc opératoire et la fabrication d'horaire d'infirmières. L'aspect stochastique a été pris en compte de manière différente en fonction des problèmes afin d'obtenir des solutions réalistes et opérationnelles.

8.1 Synthèse des travaux

Les méthodes proposées pour gérer l'incertitude sont très différentes d'un chapitre à l'autre. Leur complexité dépend fortement de la modélisation de l'incertitude.

Dans le chapitre 4, nous modélisons l'incertitude directement à travers les paramètres du problème. Les horaires d'infirmières de chaque unité sont fabriqués de manière déterministe en se basant sur des quotas d'infirmières. L'équipe volante, qui absorbe la variation de la demande en infirmières au niveau de tout l'hôpital, doit prendre un minimum en compte les prévisions du besoin global en infirmières. Les demandes en infirmières de l'équipe volante sont donc modifiées pour incorporer une notion de variance : plus la variance de la demande d'une journée est grande, plus cette demande est artificiellement augmentée.

Dans le chapitre 5, nous gérons l'incertitude de manière différente. Un ensemble de scénarios des temps d'opérations est utilisé. L'algorithme crée ensuite un emploi du temps à partir de chaque scénario. Finalement, il évalue le coût moyen de chaque emploi du temps avec une méthode de type «Monte-Carlo» et renvoie le meilleur. La distribution des temps d'opérations n'est pas prise en compte au moment de la construction de ces plannings, mais au moment de leur évaluation.

Dans le chapitre 6, nous proposons un algorithme stochastique en temps réel pour la prise de rendez-vous en radiothérapie. L'incertitude est toujours représentée à travers des scénarios de futurs patients. Cette modélisation permet d'avoir un ensemble fini de prévisions. L'algorithme utilise ensuite chacun de ces scénarios afin de trouver un ensemble de rendez-vous optimal pour tous les patients du scénario : il calcule ainsi un coût futur d'utilisation d'une plage horaire des linacs. Cette approximation agrège toute l'information stochastique dans des coûts futurs.

Dans le chapitre 7, nous formalisons et généralisons l'algorithme précédent afin de résoudre de manière stochastique et dynamique des problèmes d'allocation de ressources. L'incertitude est toujours représentée à travers des scénarios de futures requêtes d'allocation. L'algorithme utilise de deux façons les solutions du problème d'allocation obtenues avec chacun des scénarios. Premièrement, il estime le coût d'utilisation dans le futur d'une unité de chacune des ressources. Deuxièmement, il interdit les allocations qui peuvent mener à des solutions irréalisables ou non-optimales. La prise en compte de l'incertitude se fait donc totalement à travers ces deux méthodes.

8.2 Limitations de la solution proposée et améliorations futures

Pour chacune des solutions proposées, il faut trouver un compromis entre complexité du problème et précisions du modèle stochastique.

Nous présentons un algorithme très facile à implémenter dans le chapitre 4. L'algorithme utilise une heuristique classique et déterministe pour construire l'horaire de l'équipe volante. L'incertitude est juste utilisée pour modifier la demande. Des modèles plus complexes exprimant le recours pourraient être utilisés, mais ce nouveau choix se fait au détriment de la simplicité de la procédure actuelle.

Le chapitre 5 expose un algorithme utilisant une représentation de l'incertitude plus précise, mais toujours aucun recours. En effet, l'algorithme génère de manière déterministe plusieurs centaines d'emploi du temps et les évalue ensuite. Cependant, les emplois du temps optimaux d'un point de vue stochastique peuvent ne pas être construits par cette procédure, car chaque planning calculé colle parfaitement au scénario qui l'a généré. Il est peut-être nécessaire de trouver un emploi du temps intermédiaire qui s'adapte bien à chaque scénario ; un tel planning ne peut pas être fabriqué à partir d'un seul scénario, mais le peut à partir d'un ensemble. Il faut donc une modélisation du recours pour trouver ces emplois du temps optimaux.

La méthode développée dans le chapitre 6 donne de très bons résultats dans le cas particulier du CICL, car elle utilise une modélisation précise du recours. Cependant, certains aspects du problème, comme le prétraitement, ont été simplifiés et ne peuvent pas de toute façon être gérés par la procédure proposée.

En effet, le chapitre 7 montre qu'il faut ajouter un deuxième outil, soit les coupes de réalisabilité, pour résoudre le problème de prise de rendez-vous et d'ordonnancement d'un centre de radiothérapie. L'algorithme proposé utilise les solutions primales et duales des sous-problèmes pour guider globalement la procédure. Ces outils semblent résoudre un grand nombre de problèmes d'allocation de ressources. Cependant, notre procédure est complexe à implémenter, car elle utilise les décompositions de Benders et de Dantzig-Wolfe.

Finalement, les problématiques en santé doivent être résolues en collaboration avec les systèmes de santé. Il faut donc déterminer précisément, dès le début de la collaboration, les besoins et les moyens des gestionnaires pour leur proposer la meilleure solution implémentable qui corresponde à leur budget.

CHAPITRE 9 CONCLUSION

Dans cette thèse, nous avons résolu trois problèmes d'ordonnancement en santé, en tenant compte de l'incertitude qui régit le monde réel. L'objectif principal est de construire des ordonnancements flexibles qui s'adaptent bien à l'environnement réel. Nous avons proposé un cadre théorique pour modéliser des problèmes dynamiques d'allocation de ressources et un algorithme général pour résoudre ce modèle d'allocation de ressources. Cet algorithme est très efficace pour les problèmes de prise de rendez-vous. De plus, nous avons montré qu'une étude déterministe d'un problème d'ordonnancement (en particulier, l'ordonnancement des opérations d'un bloc opératoire) peut mener à des solutions très instables dans la réalité. Enfin, nous avons illustré qu'il fallait trouver un équilibre entre la précision du modèle d'incertitude et la complexité de l'algorithme. Bien qu'une caractérisation plus fine de l'incertitude permette d'obtenir de meilleures solutions, elle complexifie la transférabilité des algorithmes aux organisations partenaires. L'important est donc de bien définir les problèmes que l'on souhaite résoudre.

«On résout les problèmes qu'on se pose et non les problèmes qui se posent.»
– Henri Poincaré

Cette thèse est le fruit de quatre années de travail qui m'ont permis de lancer plein de projets et de m'intéresser à de multiples problèmes. J'espère que ma thèse n'est finalement que le début d'une carrière académique longue et prolifique. C'est en tout cas ainsi que j'ai décidé d'orienter ma vie professionnelle. Encore merci à tous ceux qui m'ont permis d'aller dans cette direction.

RÉFÉRENCES

"Décès selon les principaux groupes de causes, sexes réunis, québec, 2001-2011", Institut de la statistique du Québec, Rapp. tech., 2012.

U. Aickelin et K. A. Dowsland, "An indirect genetic algorithm for a nurse scheduling problem", *Computers & Operations Research*, vol. 31, pp. 761–778, 2004.

U. Aickelin, E. K. Burke, et J. Li, "An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering", *Journal of the Operational Research Society*, vol. 58, pp. 1574–1585, 2007.

AIMMS-MOPTA 2013 Challenge, "Operation room management under uncertainty", `http://coral.ie.lehigh.edu/~mopta2013/competition`, 2013, Accessed : 2015-07-28.

A. N. Avramidis, A. Deslauriers, et P. L'Ecuyer, "Modeling daily arrivals to a telephone call center", *Management Science*, vol. 50, no. 7, pp. 896–908, 2004.

Banque mondiale, "Dépenses en santé, total (% du Produit Intérieur Brut)", `http://donnees.banquemondiale.org/indicateur/SH.XPD.TOTL.ZS/countries/1W-US-CA-FR?display=graph`, 2013, Accédé : 08-10-2015.

J. F. Bard et H. W. Purnomo, "Hospital-wide reactive scheduling of nurses with preference considerations", *IIE Transactions*, vol. 37, pp. 589–608, 2005.

——, "Short-term nurse scheduling in response to daily fluctuations in supply and demand", *Health Care Management Science*, vol. 31, pp. 315–324, 2005.

——, "Cyclic preference scheduling of nurses using a lagrangian-based heuristic", *Journal of Scheduling*, vol. 10, pp. 5–23, 2007.

J. F. Bard et L. Wan, "The task assignment problem for unrestricted movement between workstation groups", *Journal of Scheduling*, vol. 9, no. 4, pp. 315–341, 2006.

S. Batun et M. A. Begen, "Optimization in healthcare delivery modeling : Methods and applications", dans *Handbook of Healthcare Operations Management*. Springer, 2013, pp. 75–119.

N. Bäuerle et U. Rieder, *Markov Decision Processes with Applications to Finance*. Springer Science & Business Media, 2011.

M. A. Begen et M. Queyranne, “Appointment scheduling with discrete random durations”, *Mathematics of Operations Research*, vol. 36, no. 2, pp. 240–257, 2011.

J. Beliën et E. Demeulemeester, “A branch-and-price approach for integrating nurse and surgery scheduling”, *European Journal of Operational Research*, vol. 189, no. 3, pp. 652–668, 2008. DOI : 10.1016/j.ejor.2006.10.060

J. F. Benders, “Partitioning procedures for solving mixed-variables programming problems”, *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

B. Berg, B. Denton, H. Nelson, H. Balasubramanian, A. Rahman, A. Bailey, et K. Lindor, “A discrete event simulation model to evaluate operational performance of a colonoscopy suite”, *Medical Decision Making*, vol. 30, no. 3, pp. 380–387, 2009. DOI : 10.1177/0272989X09345890

I. Berrada, J. A. Ferland, et P. Michelon, “A multi-objective approach to nurse scheduling with both hard and soft constraints”, *Socio-Economic Planning Sciences*, vol. 30, pp. 183–193, 1996.

S. Bertel et J.-C. Billaut, “A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation”, *European Journal of Operational Research*, vol. 159, no. 3, pp. 651–662, 2004.

D. Bertsimas et M. Sim, “Robust discrete optimization and network flows”, *Mathematical programming*, vol. 98, no. 1-3, pp. 49–71, 2003.

J. R. Birge et F. Louveaux, *Introduction to Stochastic Programming*, 2e éd., série Springer Series in Operations Research and Financial Engineering. New York, NY, USA : Springer, 2011. DOI : 10.1007/978-1-4614-0237-4

Z. I. Botev, P. L’Ecuyer, et B. Tuffin, “Markov chain importance sampling with applications to rare event probability estimation”, *Statistics and Computing*, pp. 1–15, 2011.

P. Boucher et H. Beauregard, “Le coût économique du cancer au Québec, en 2008”, Coalition priorité cancer au Québec, Rapp. tech., 2010.

V. Boyer, B. Gendron, et L.-M. Rousseau, “A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem”, *Journal of Scheduling*, vol. 17, no. 2, pp. 185–197, 2014.

N. Buchbinder, “Designing competitive online algorithms via a primal-dual approach”, Thèse de doctorat, Technion, 2008.

E. K. Burke, P. De Causmaecker, S. Petrovic, et G. Vanden Berghe, “Metaheuristics for handling time interval coverage constraints in nurse scheduling”, *Applied Artificial Intelligence*, vol. 20, pp. 743–766, 2006.

E. K. Burke, T. Curtois, R. Qu, et G. Vanden Berghe, “A scatter search methodology for the nurse rostering problem”, *Journal of the Operational Research Society*, vol. 61, pp. 1667–1679, 2010.

E. K. Burke, P. De Causmaecker, et G. Vanden Berghe, “Novel meta-heuristic approaches to nurse rostering problems in belgian hospitals”, *Handbook of scheduling : algorithms, models and performance analysis*, vol. 18, pp. 1–44, 2004.

Canadian Cancer Society’s Advisory Committee on Cancer Statistics, “Canadian cancer statistics 2013”, Canadian Cancer Society, Toronto, ON, Rapp. tech., 2013.

B. Cardoen, E. Demeulemeester, et J. Beliën, “Operating room planning and scheduling : A literature review”, *European Journal of Operational Research*, vol. 201, no. 3, pp. 921–932, 2010. DOI : `10.1016/j.ejor.2009.04.011`

T. Cayirli et E. Veral, “Outpatient scheduling in health care : a review of literature”, *Production and Operations Management*, vol. 12, no. 4, pp. 519–549, 2003.

M. T. Cezik et P. L’Ecuyer, “Staffing multiskill call centers via linear programming and simulation”, *Management Science*, vol. 54, no. 2, pp. 310–323, 2008.

N. Channouf et P. L’Ecuyer, “A normal copula model for the arrival process in a call center”, *International Transactions in Operational Research*, 2012.

Z. Chen, W. King, R. Pearcey, M. Kerba, et W. J. Mackillop, “The relationship between waiting time for radiotherapy and clinical outcomes : A systematic review of the literature”, *Radiotherapy and Oncology*, vol. 87, no. 1, pp. 3–16, 2008.

D. F. Ciocan et V. F. Farias, “Dynamic allocation problems with volatile demand”, *Mathematics of Operations Research*, vol. 37, no. 3, pp. 501–525, 2012.

A. R. Clark et H. Walker, “Nurse rescheduling with shift preferences and minimal disruption”, *Journal of Applied Operational Research*, vol. 3, pp. 148–162, 2011.

Comité consultatif de la Société canadienne du cancer, “Statistiques canadiennes sur le cancer”, Société canadienne du cancer, Toronto, ON, Rapp. tech., 2013.

D. Conforti, F. Guerriero, et R. Guido, “Optimization models for radiotherapy patient scheduling”, *4OR : A Quarterly Journal of Operations Research*, vol. 6, no. 3, pp. 263–278, 2008.

——, “Non-block scheduling with priority for radiotherapy treatments”, *European Journal of Operational Research*, vol. 201, no. 1, pp. 289–296, 2009.

A. Corominas, R. Pastor, et E. Rodríguez, “Rotational allocation of tasks to multifunctional workers in a service industry”, *International Journal of Production Economics*, vol. 103, no. 1, pp. 3–9, 2006.

G. B. Dantzig et P. Wolfe, “Decomposition principle for linear programs”, *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

E. Demeulemeester, J. Beliën, B. Cardoen, et M. Samudra, “Operating room planning and scheduling”, dans *Handbook of Healthcare Operations Management.* Springer, 2013, pp. 121–152. DOI : 10.1007/978-1-4614-5885-2_5

B. Denton et D. Gupta, “A sequential bounding approach for optimal appointment scheduling”, *IIE Transactions*, vol. 35, no. 11, pp. 1003–1016, 2003. DOI : 10.1080/07408170304395

B. Denton, J. Viapiano, et A. Vogl, “Optimization of surgery sequencing and scheduling decisions under uncertainty”, *Health care management science*, vol. 10, no. 1, pp. 13–24, 2007. DOI : 10.1007/s10729-006-9005-4

G. Desaulniers, J. Desrosiers, et M. M. Solomon, *Column Generation.* Springer, 2005.

F. Dexter et R. D. Traub, “How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time”, *Anesthesia & Analgesia*, vol. 94, no. 4, pp. 933–942, 2002. DOI : 10.1097/00000539-200204000-00030

F. Dexter, A. Macario, et R. D. Traub, “Which algorithm for scheduling add-on elective cases maximizes operating room utilization ? : Use of bin packing algorithms and fuzzy constraints in operating room management”, *Anesthesiology*, vol. 91, no. 5, pp. 1491–1500, 1999. DOI : 10.1097/00000542-199911000-00043

K. A. Dowsland et J. M. Thompson, “Solving a nurse scheduling problem with knapsacks, networks and tabu search”, *Journal of the Operational Research Society*, vol. 51, pp. 825–833, 2000.

S. A. Erdogan et B. Denton, “Dynamic appointment scheduling of a stochastic server with uncertain demand”, *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 116–132, 2013.

S. A. Erdogan, A. Gose, et B. T. Denton, “On-line appointment sequencing and scheduling”, *IIE Transactions*, no. just-accepted, pp. 00–00, 2015.

A. T. Ernst, H. Jiang, M. Krishnamoorthy, et D. Sier, “Staff scheduling and rostering : A review of applications, methods and models”, *European Journal of Operational Research*, vol. 153, no. 1, pp. 3–27, 2004.

J. Feldman, A. Mehta, V. Mirrokni, et S. Muthukrishnan, “Online stochastic matching : Beating 1-1/e”, dans *50th Annual IEEE Symposium on Foundations of Computer Science, 2009 (FOCS’09)*. IEEE, 2009, pp. 117–126.

J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, et C. Stein, “Online stochastic packing applied to display ad allocation”, dans *Algorithms–ESA 2010*. Springer, 2010, pp. 182–194.

J. A. Ferland, I. Berrada, I. Nabli, B. Ahiod, P. Michelon, V. Gascon, et E. Gagné, “Generalized assignment type goal programming problem : Application to nurse scheduling”, *Journal of Heuristics*, vol. 7, pp. 391–413, 2001.

S. Gélinas et F. Soumis, *Dantzig-Wolfe Decomposition for Job Shop Scheduling*. Springer, 2005.

M. Gendreau, G. Laporte, et R. Séguin, “Stochastic vehicle routing”, *European Journal of Operational Research*, vol. 88, no. 1, pp. 3 – 12, 1996. DOI : `10.1016/0377-2217(95)00050-X`

F. Glover et M. Laguna, *Tabu Search*. Kluwer Academic Publishers : Boston., 1997.

J. Gu, M. Goetschalckx, et L. F. McGinnis, “Research on warehouse operation : A comprehensive review”, *European Journal of Operational Research*, vol. 177, no. 1, pp. 1–21, 2007.

F. Guerriero et R. Guido, “Operational research in the management of the operating theatre : a survey”, *Health care management science*, vol. 14, no. 1, pp. 89–114, 2011. DOI : `10.1007/s10729-010-9143-6`

A. Guinet et S. Chaabane, “Operating theatre planning”, *International Journal of Production Economics*, vol. 85, no. 1, pp. 69–81, 2003. DOI : 10.1016/S0925-5273(03)00087-2

D. Gupta et B. Denton, “Appointment scheduling in health care : Challenges and opportunities”, *IIE transactions*, vol. 40, no. 9, pp. 800–819, 2008.

A. Hanset, N. Meskens, et D. Duvivier, “Using constraint programming to schedule an operating theatre”, dans *Health Care Management (WHCM), 2010 IEEE Workshop on*. IEEE, 2010, pp. 1–6. DOI : 10.1109/WHCM.2010.5441245

P. R. Harper, “A Framework for Operational Modelling of Hospital Resources”, *Health Care Management Science*, vol. 5, no. 3, pp. 165–173, 2002. DOI : 10.1023/A:1019767900627

Health Council Canada, “A 10-year plan to strengthen health care”, Rapp. tech., 2004.

T. R. Huschka, B. Denton, S. Gul, et J. W. Fowler, “Bi-criteria evaluation of an outpatient procedure center via simulation”, dans *2007 Winter Simulation Conference*. IEEE, December 2007, pp. 1510–1518. DOI : 10.1109/WSC.2007.4419764

IBM. (2014) Ilog cplex optimization studio. En ligne : http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/

Inauguration du centre intégré de cancérologie de Montérégie, “http://www.premier-ministre.gouv.qc.ca/actualites/communiques/details.asp?idCommunique=222”, Rapp. tech., 2011.

Y. Jacquemin, E. Marcon, et P. Pommier, “Towards an improved resolution of radiotherapy, scheduling”, dans *2010 IEEE Workshop on Health Care Management (WHCM)*, 2010, pp. 1–6.

P. Jaillet et X. Lu, “Near-optimal online algorithms for dynamic resource allocation problems”, *arXiv preprint arXiv :1208.2596*, 2012.

——, “Online stochastic matching : New algorithms with better bounds”, *Mathematics of Operations Research*, vol. 39, no. 3, pp. 624–646, 2014.

P. Jaillet et M. R. Wagner, “Online optimization-an introduction”, *TutORials in Operations Research : Risk and Optimization in an Uncertain World, INFORMS*, pp. 142–152, 2010.

T. Kapamara, K. Sheibani, O. Haas, C. Reeves, et D. Petrovic, “A review of scheduling problems in radiotherapy”, dans *Proceedings of the 18th International Conference on Systems*

Engineering (ICSE2006), Coventry University, UK, 2006, pp. 201–207.

C. Karande, A. Mehta, et P. Tripathi, “Online bipartite matching with unknown distributions”, dans *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing.* ACM, 2011, pp. 587–596.

R. M. Karp, U. V. Vazirani, et V. V. Vazirani, “An optimal algorithm for on-line bipartite matching”, dans *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing.* ACM, 1990, pp. 352–358.

D. L. Kellogg et S. Walczak, “Nurse scheduling : from academia to implementation or not ?” *Interfaces*, vol. 37, no. 4, pp. 355–369, 2007.

K. J. Klassen et T. R. Rohleder, “Outpatient appointment scheduling with urgent clients in a dynamic, multi-period environment”, *International Journal of Service Industry Management*, vol. 15, no. 2, pp. 167–186, 2004. DOI : 10.1108/09564230410532493

A. J. Kleywegt, A. Shapiro, et T. Homem-de Mello, “The sample average approximation method for stochastic discrete optimization”, *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002. DOI : 10.1137/S1052623499363220

A. Legrain et P. Jaillet, *Stochastic Online Bipartite Resource Allocation Problems.* CIRRELT, 2013.

A. Legrain, M.-A. Fortin, N. Lahrichi, et L.-M. Rousseau, “Online stochastic optimization of radiotherapy patient scheduling”, *Health Care Management Science*, vol. 3, no. 3, pp. 1–14, 2014.

A. Legrain, M.-A. Fortin, N. Lahrichi, L.-M. Rousseau, et M. Widmer, “Stochastic optimization of the scheduling of a radiotherapy center”, dans *Journal of Physics : Conference Series*, vol. 616, no. 1. IOP Publishing, 2015, p. 012008.

B. Maenhout et M. Vanhoucke, “An electromagnetic meta-heuristic for the nurse scheduling problem”, *Journal of Heuristics*, vol. 13, pp. 359–385, 2007.

——, “New computational results for the nurse scheduling problem : a scatter search algorithm”, dans *Evolutionary Computation in Combinatorial Optimization.* Springer, 2006, pp. 159–170.

V. H. Manshadi, S. O. Gharan, et A. Saberi, “Online stochastic matching : Online actions based on offline statistics”, *Mathematics of Operations Research*, vol. 37, no. 4, pp. 559–573, 2012.

A. Mehta, A. Saberi, U. Vazirani, et V. Vazirani, “Adwords and generalized online matching”, *Journal of the ACM (JACM)*, vol. 54, no. 5, 2007.

D. Min et Y. Yih, “Scheduling elective surgery under uncertainty and downstream capacity constraints”, *European Journal of Operational Research*, vol. 206, no. 3, pp. 642 – 652, 2010. DOI : 10.1016/j.ejor.2010.03.014

Ministère de la santé et des services sociaux, “Plan stratégique 2010–2015”, Rapp. tech., 2010.

Ministère de la Santé et des Services Sociaux du Québec, “Msss - québec - montants des dépenses consacrées à la santé et aux services sociaux, selon les éléments de la structure budgétaire, au québec, 2000-2001 à 2013-2014”, http://www.informa.msss.gouv.qc.ca/Details.aspx?Id=xwwbq/us7X8=, 2015, Accédé : 08-10-2015.

R. M’Hallah et A. Al-Roomi, “The planning and scheduling of operating rooms : A simulation approach”, *Computers & Industrial Engineering*, vol. 78, pp. 235–248, 2014. DOI : 10.1016/j.cie.2014.07.022

C. Novoa et R. Storer, “An approximate dynamic programming approach for the vehicle routing problem with stochastic demands”, *European Journal of Operational Research*, vol. 196, no. 2, pp. 509–515, 2009.

Ordre des infirmières et infirmiers auxiliaires du Québec, “Salles d’opération : caractéristiques, évolution et utilisation des infirmières auxiliaires”, http://www.oiiaq.org/documents/file/rapport-salles-operation.pdf, 2008, Accédé : 08-10-2015.

Ordre des infirmières et infirmiers du Québec, “Le québec compte 8,3% de plus d’infirmières et d’infirmiers qu’il y a dix ans”, http://www.oiiq.org/uploads/periodiques/infostats/vol06no01/index.htm, 2014, Accédé : 08-10-2015.

J. Patrick, M. L. Puterman, et M. Queyranne, “Dynamic multipriority patient scheduling for a diagnostic resource”, *Operations Research*, vol. 56, no. 6, pp. 1507–1525, 2008.

——, “Reducing wait times through operations research : Optimizing the use of surge capacity”, *Healthcare Policy*, vol. 3, no. 3, pp. 75–88, 2008.

D. Petrovic, M. Morshed, et S. Petrovic, “Genetic algorithm based scheduling of radiotherapy treatments for cancer patients”, dans *Artificial Intelligence in Medicine*, série Lecture

Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5651, pp. 101–105.

S. Petrovic et P. Leite-Rocha, “Constructive and GRASP approaches to radiotherapy treatment scheduling”, dans *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science*, 2008, pp. 192–200.

S. Petrovic, W. Leung, X. Song, et S. Sundar, “Algorithms for radiotherapy treatment booking”, dans *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG'2006), Nottingham, UK*, 2006.

D.-N. Pham et A. Klinkert, “Surgical case scheduling as a generalized job shop scheduling problem”, *European Journal of Operational Research*, vol. 185, no. 3, pp. 1011–1025, 2008. DOI : `10.1016/j.ejor.2006.03.059`

W. B. Powell, *Approximate Dynamic Programming : Solving the Curses of Dimensionality*. John Wiley & Sons, 2007.

W. B. Powell et B. V. Roy, “Approximate dynamic programming for high dimensional resource allocation problems”, dans *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press, Los Alamitos, CA, 2004, pp. 261–280.

P. Punnakitikashem, J. M. Rosenberger, et D. B. Behan, “Stochastic programming for nurse assignment”, *Computational Optimization and Applications*, vol. 40, no. 3, pp. 321–349, 2008.

M. L. Puterman, *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

J. I. Rubrico, T. Higashi, H. Tamura, et J. Ota, “Online rescheduling of multiple picking agents for warehouse management”, *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 62–71, 2011.

R. Ruiz et J. A. Vázquez-Rodríguez, “The hybrid flow shop scheduling problem”, *European Journal of Operational Research*, vol. 205, no. 1, pp. 1–18, 2010.

H. Saadouli, B. Jerbi, A. Dammak, L. Masmoudi, et A. Bouaziz, “A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department”, *Computers & Industrial Engineering*, vol. 80, pp. 72–79, 2015. DOI : `10.1016/j.cie.2014.11.021`

P. Santibáñez, M. Begen, et D. Atkins, "Surgical block scheduling in a system of hospitals : an application to resource and wait list management in a british columbia health authority", *Health care management science*, vol. 10, no. 3, pp. 269–282, 2007.

A. Sauré, J. Patrick, S. Tyldesley, et M. L. Puterman, "Dynamic multi-appointment patient scheduling for radiation therapy", *European Journal of Operational Research*, vol. 223, no. 2, pp. 573–584, 2012.

D. Sier, P. Tobin, et C. McGurk, "Scheduling surgical procedures", *The Journal of the Operational Research Society*, vol. 48, no. 9, pp. 884–891, 1997. DOI : `10.1057/palgrave.jors.2600441`

S. P. Siferd et W. Benton, "A decision modes for shift scheduling of nurses", *European Journal of Operational Research*, vol. 74, no. 3, pp. 519–527, 1994.

R. V. Slyke et R. J. Wets, "L-shaped linear programs with application to optimal control and stochastic programming", *SIAM Journal and Applied Mathematics*, pp. 638–663, 1969.

C. Valouxis et E. Housos, "Hybrid optimization techniques for the workshift and rest assignment of nursing personnel", *Artificial intelligence in Medicine*, vol. 20, pp. 155–175, 2000.

J. Van den Bergh, B. Jeroen, D. Bruecker Philippe, D. Erik, et D. Boeck Liesje, "Personnel scheduling : A literature review", *European Journal of Operational Research*, vol. 226, pp. 367–385, 2013.

P. Van Hentenryck et R. Bent, *Online Stochastic Combinatorial Optimization*. The MIT Press, 2009.

M. Van Houdenhoven, J. M. van Oostrum, E. W. Hans, G. Wullink, et G. Kazemier, "Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling", *Anesthesia & Analgesia*, vol. 105, no. 3, pp. 707–714, 2007. DOI : `10.1213/01.ane.0000277492.90805.0f`

J. M. Welton, M. H. Fisher, S. DeGrace, et L. Zone-Smith, "Hospital nursing costs, billing, and reimbursement", *Nursing Economics*, vol. 24, pp. 239–45, 2006.

T. Wong, M. Xu, et K. Chin, "A two-stage heuristic approach for nurse scheduling problem : A case study in an emergency department", *Computers & Operations Research*, 2014.

World Bank, “Health expenditure, total (% of Gross Domestic Product)”, http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS/countries/1W-US?display=graph, 2011, Accessed : 2014-02-18.

E. Yilmaz, “A mathematical programming model for scheduling of nurses’ labor shifts”, *Journal of medical systems*, vol. 36, no. 2, pp. 491–496, 2012.